The Contextual Loss for Image Transformation with Non-Aligned Data

Related Works

- Image transformation problems rely on loss functions that measure the similarity between the generated image and a target image.

Fig. 10. Puppet control: Results of animating a "puppet" (Ray Kurzweil) according

- The commonly used loss functions for comparing images can be classified into two types:
  - (i) Pixel-to-pixel loss functions that compare pixels at the same spatial coordinates, e.g., (L2, L1, and the perceptual loss).
  - (ii) Global loss functions, such as the Gram loss, which successfully captures style and texture by comparing statistics collected over the entire image.

Other loss functions: In the following we compare the Contextual loss to other popular loss functions. We provide here their definitions for completeness:

- The Perceptual loss [8] $\mathcal{L}_P(x, y, l_P) = ||\Phi^{l_P}(x) - \Phi^{l_P}(y)||_1$, where Φ is VGG19 [31] and l_P represents the layer.
- The $L1$ loss $\mathcal{L}_1(x, y) = ||x - y||_1$.
- The $L2$ loss $\mathcal{L}_2(x, y) = ||x - y||_2$.
- The Gram loss [6] $\mathcal{L}_{Gram}(x, y, l_\mathcal{G}) = ||\mathcal{G}_\Phi^{l_\mathcal{G}}(x) - \mathcal{G}_\Phi^{l_\mathcal{G}}(y)||_F^2$, where the Gram matrices $\mathcal{G}_\Phi^{l_\mathcal{G}}$ of layer $l_\mathcal{G}$ of Φ are as defined in [6].

Motivation

- The commonly used loss functions for comparing images can be classified into two types:
  - (i) Pixel-to-pixel loss functions
    - Not designed for problems where the training data is not aligned.
  - (ii) Global loss functions, such as the Gram loss, which successfully captures style and texture by comparing statistics collected over the entire image.
    - Due to its global nature it translates global characteristics to the entire image.

- Toy experiment with non-aligned data

(a) Noisy input (b) Clean targets (c) $\mathcal{L}_1$ as loss (d) $\mathcal{L}_{\mathrm{CX}}$ as loss

Fig. 6. Robustness to misalignments: A noisy input image (a) is cleaned via gradient descent, where the target clean images (b) show the same scene, but are not aligned with the input. Optimizing with $\mathcal{L}_1$ leads to a highly blurred result (c) while optimizing with our contextual loss $\mathcal{L}_{\mathrm{CX}}$ removes the noise nicely (d). This is since $\mathcal{L}_{\mathrm{CX}}$ is robust to misalignments and spatial deformations.

- Aligned training pairs of images will not be available.
  - To overcome the lack of training pairs the simple feed-forward architectures were replaced with more complex ones, e.g., CycleGAN requires four different networks.

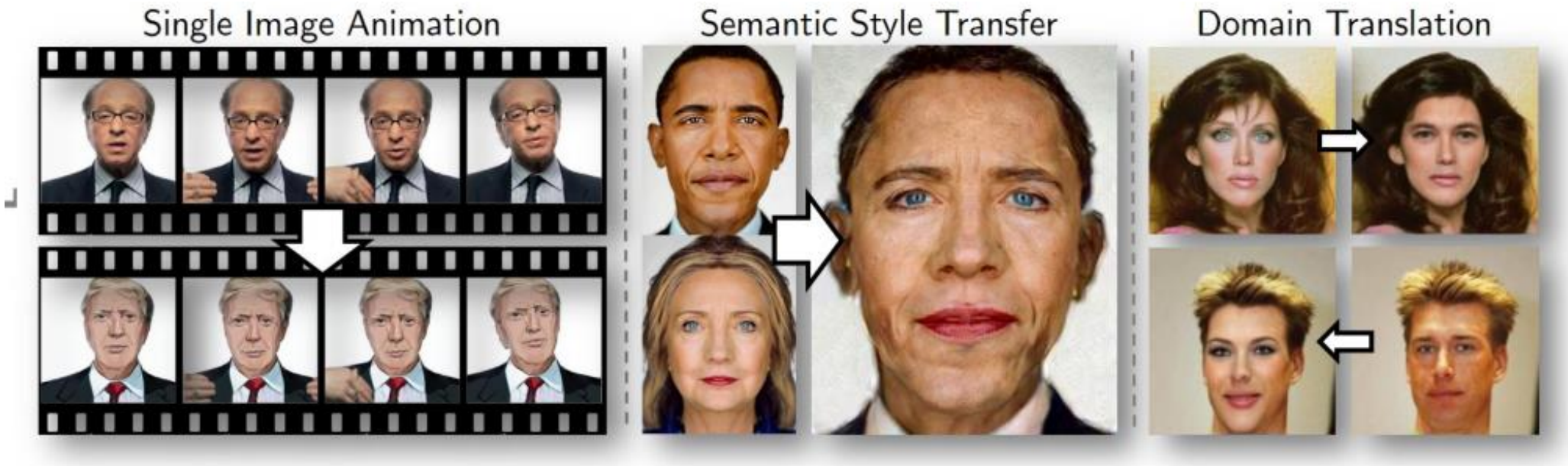
Fig. 11. Unpaired domain transfer: Gender transformation with unpaired data

Contribution

- We present an alternative **loss function** that does not require alignment.

- Our loss is based on both context and semantics – it compares regions with similar semantic meaning, while considering the context of the entire image.

- The Contextual loss provides a simple solution to all of these tasks.

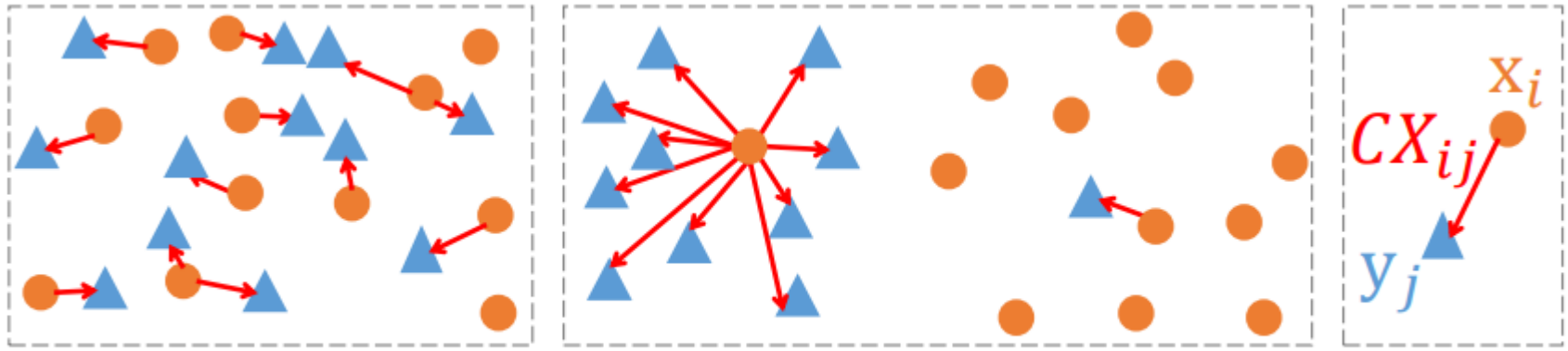


Method

- Given an image x and a target image y we represent each as a collection of points $X = \{x_i\}$ and $Y = \{y_j\}$. We assume $|X| = |Y| = N$.

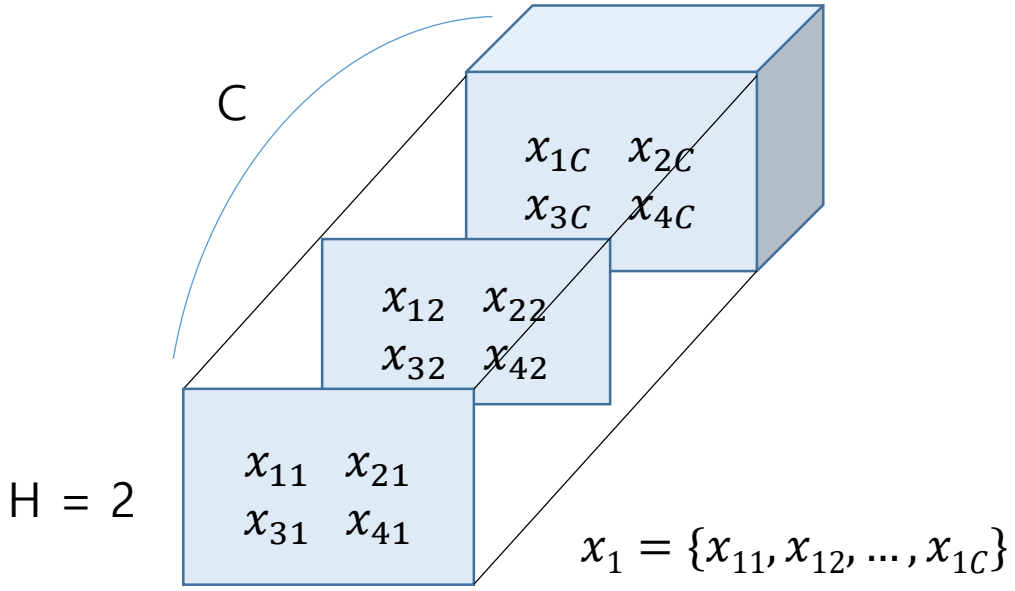
- We consider feature x_i as **contextually similar** to feature y_j if it is significantly **closer** to it than to all other features in Y.

- we consider a pair of images as similar when <u>for most features</u> of one image there exist similar features in the other.

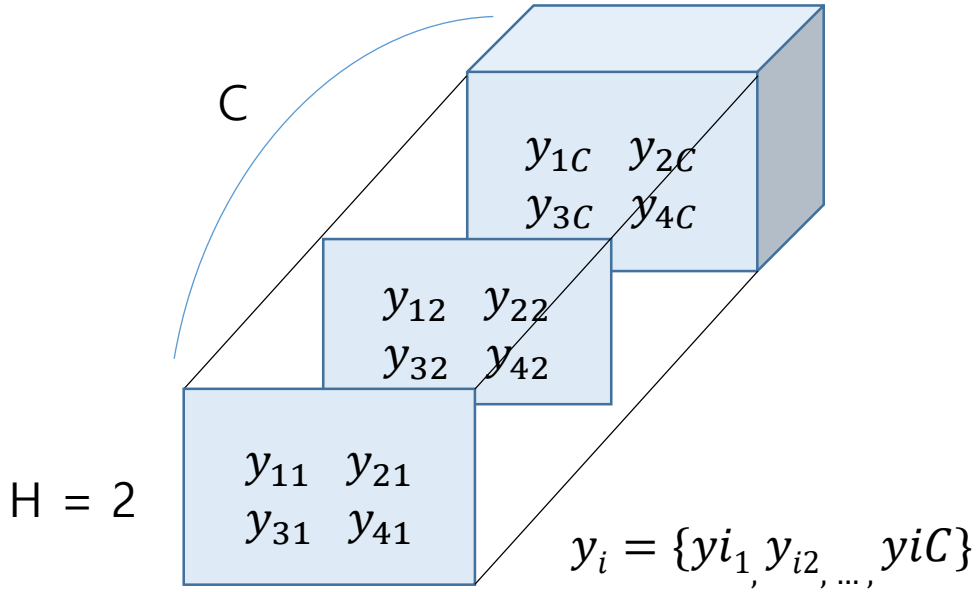


(a) Similar (b) Not-similar

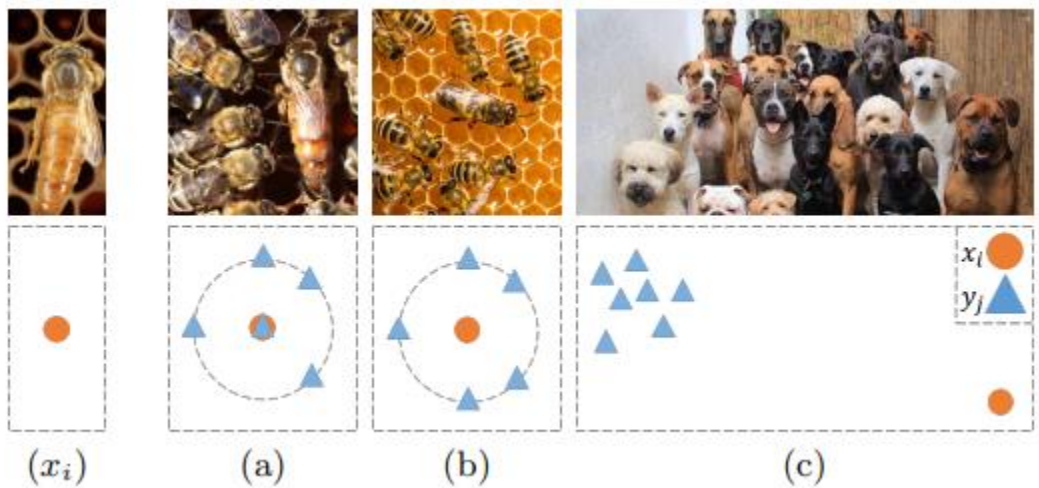
- We extract the corresponding set of features from the images by passing them through a perceptual, VGG19.



$X = \{x_1, x_2, x_3, x_4\}$

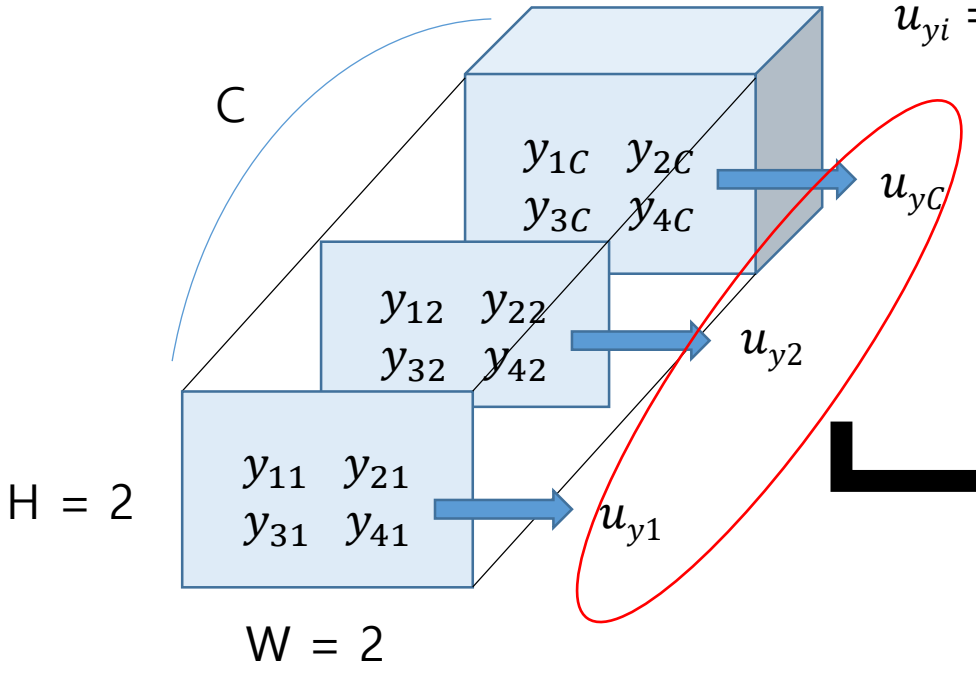


$Y = \{y_1, y_2, y_3, y_4\}$



$$d_{ij} = (1 - \frac{(x_i - \mu_y) \cdot (y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N} \sum_j y_j.$$

- Normalizing features
  - 1) Compute u_y



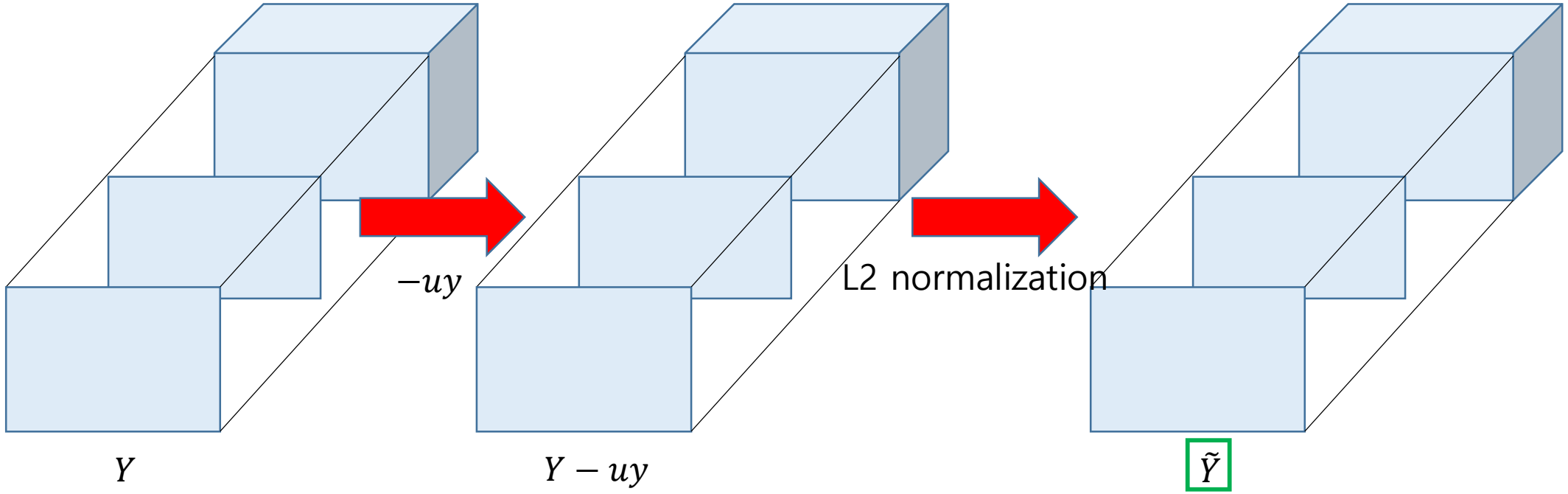
$$u_{yi} = \frac{(y_{1i} + y_{2i} + y_{3i} + y_{4i})}{4}$$

$$d_{ij} = (1 - \frac{(x_i - \mu_y)\cdot(y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N}\sum_j y_j.$$

$$u_y = \{uy_1, u_{y2}, \ldots, uyC\}$$

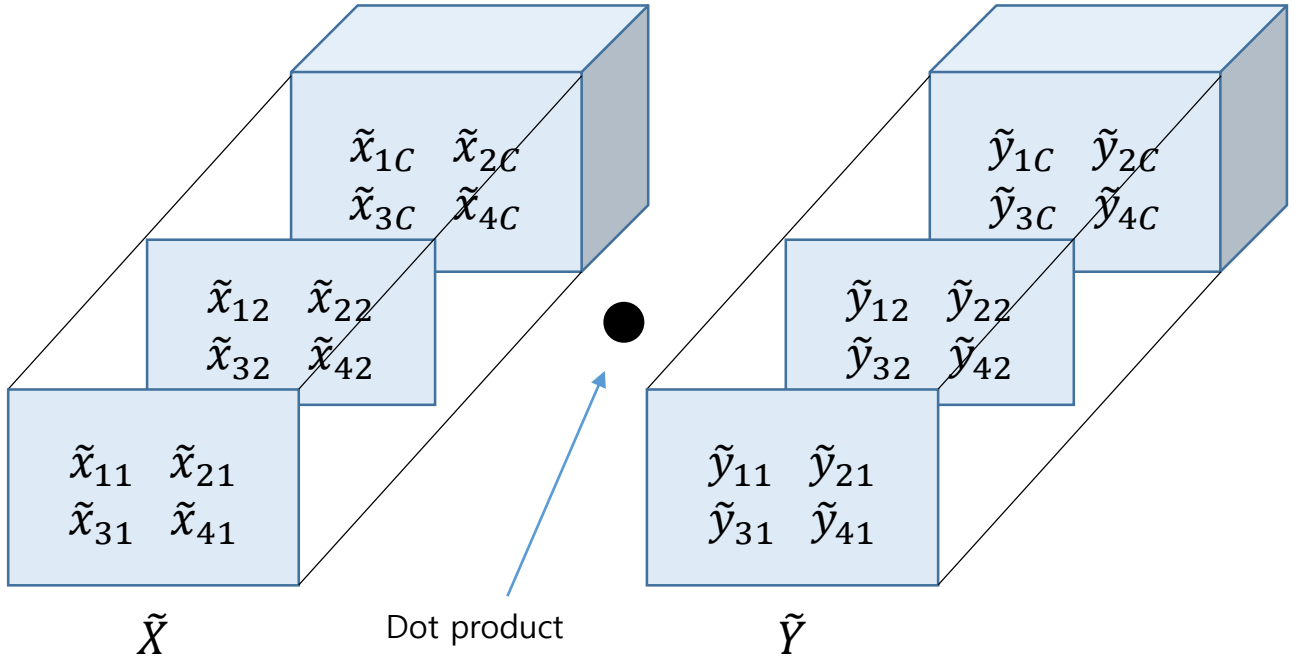
$$Y = \{y_1, y_2, y_3, y_4\}$$

- Normalizing features
  - 1) Compute u_y
  - 2) L2 normalization



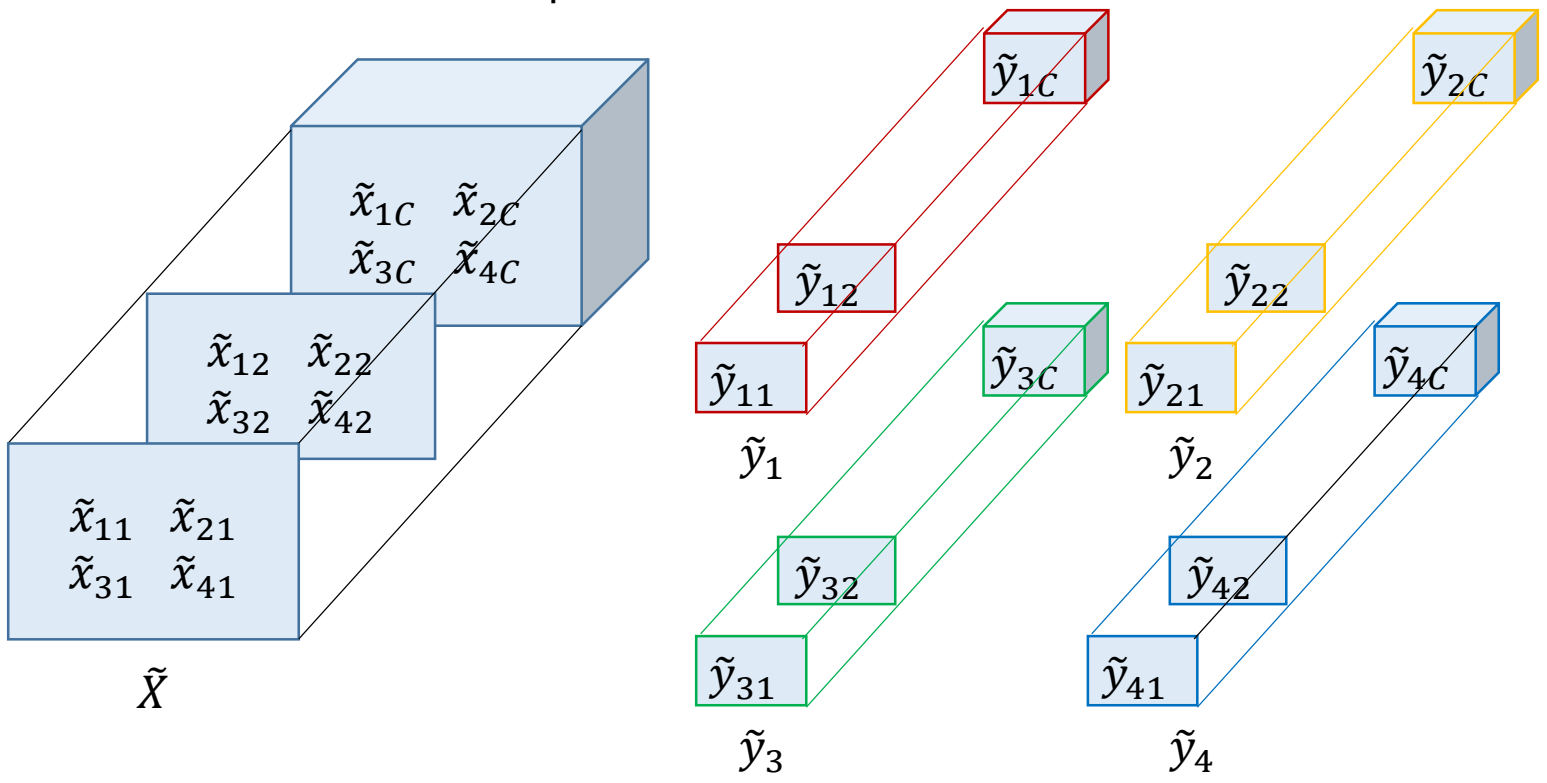
$$d_{ij} = (1 - \frac{(x_i - \mu_y)(y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N} \sum_j y_j.$$

- Distance from normalized features
  - To compute distance, Do dot product



$$d_{ij} = (1 - \frac{(x_i - \mu_y)\cdot(y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N}\sum_j y_j.$$

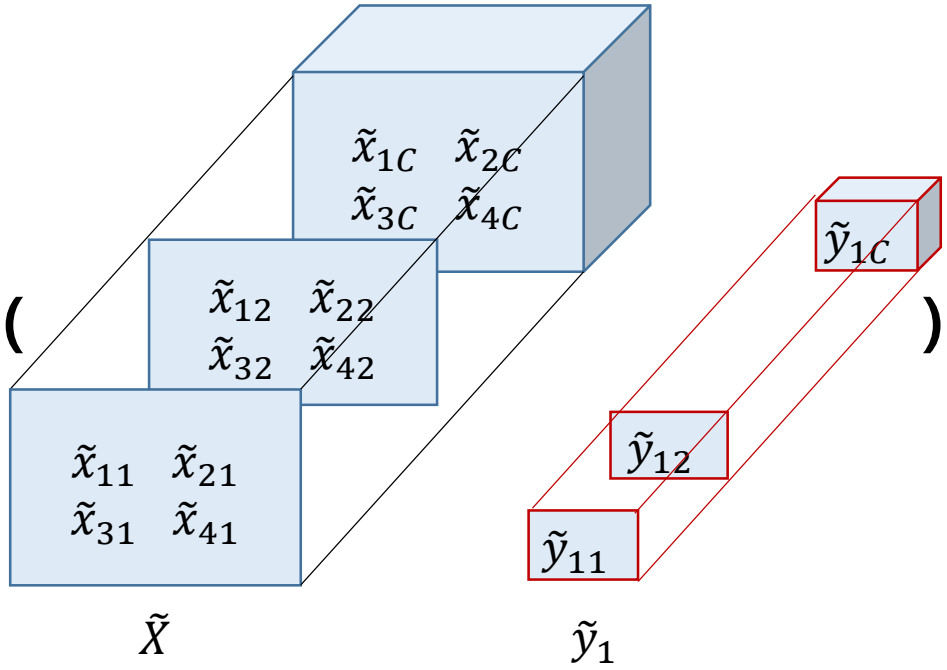
- Distance from normalized features
  - To compute distance, Do dot product
  - We can use convolution operation



$$d_{ij} = (1 - \frac{(x_i - \mu_y)\cdot(y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N}\sum_j y_j.$$

- Distance from normalized features
  - To compute distance, Do dot product
  - We can use convolution operation

Conv2d(



$\tilde{X}$ $\tilde{y}_1$

)

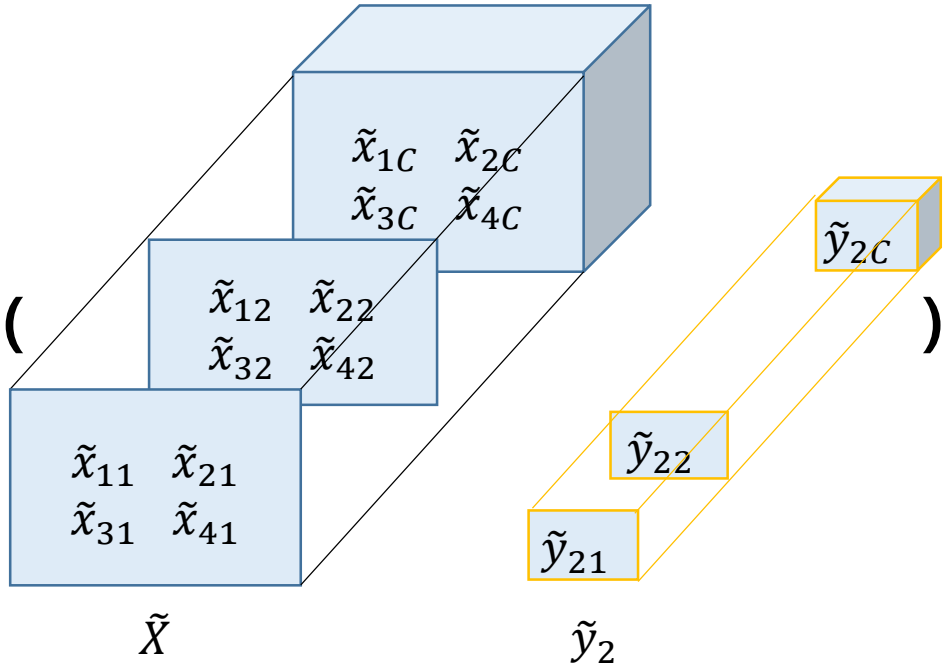
$\tilde{x}_1 \cdot \tilde{y}_1$ $\tilde{x}_2 \cdot \tilde{y}_1$
$\tilde{x}_3 \cdot \tilde{y}_1$ $\tilde{x}_4 \cdot \tilde{y}_1$

$\tilde{X} \cdot \tilde{y}_1$

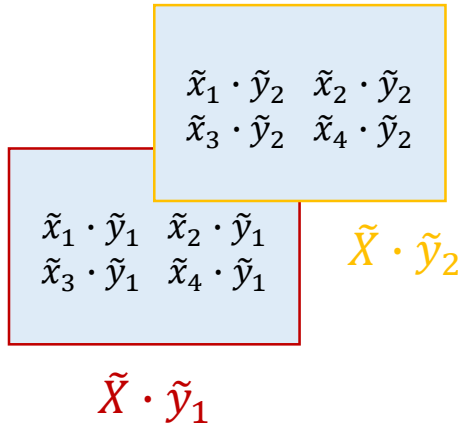
$$d_{ij} = (1 - \frac{(x_i - \mu_y) \cdot (y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N} \sum_j y_j.$$

- Distance from normalized features
  - To compute distance, Do dot product
  - We can use convolution operation

Conv2d(

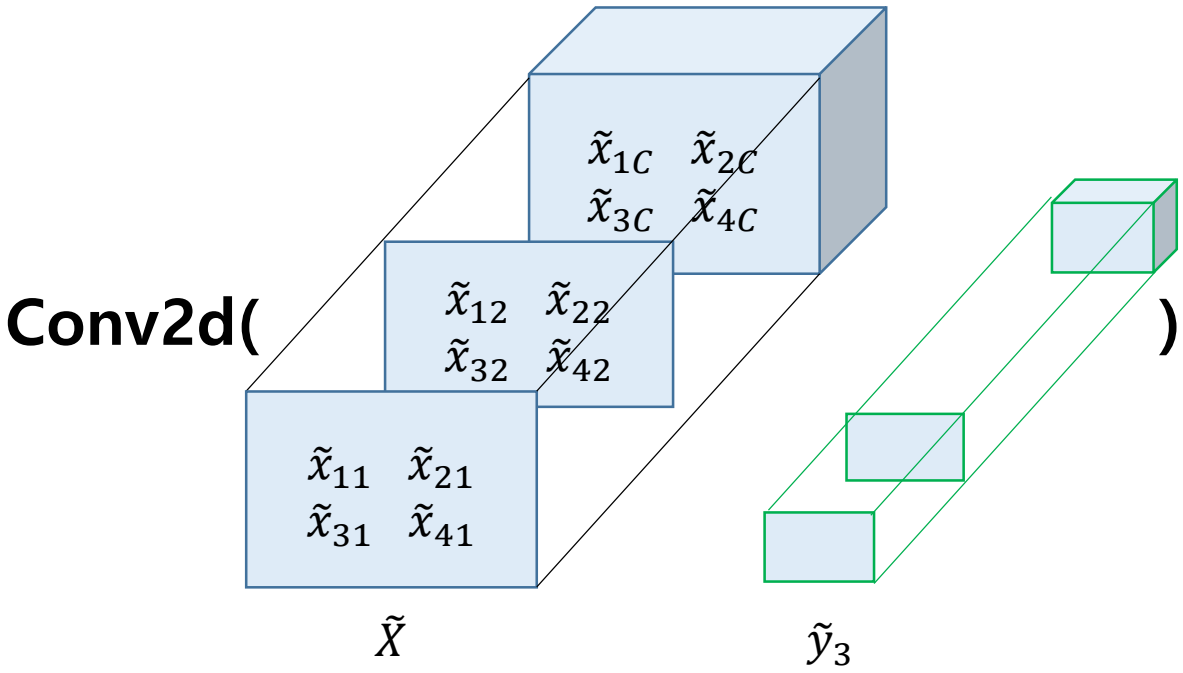


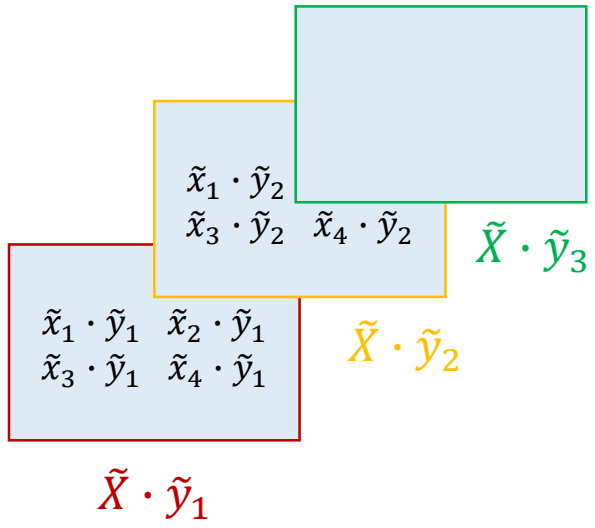
)



$$d_{ij} = (1 - \frac{(x_i - \mu_y) \cdot (y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N} \sum_j y_j.$$

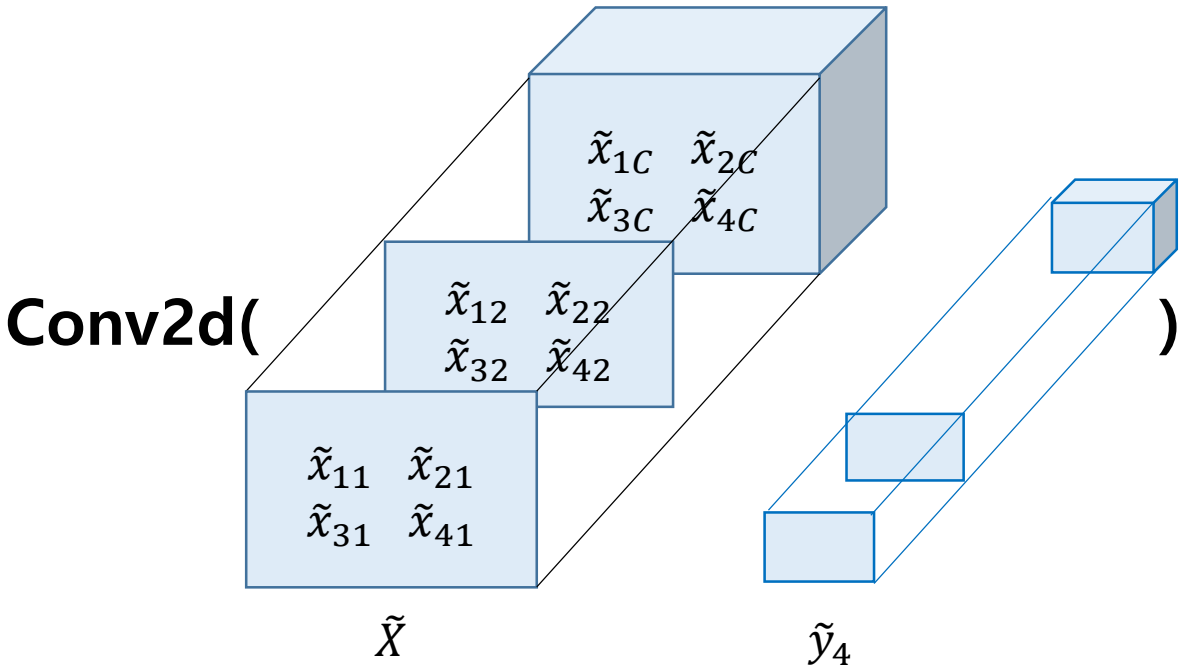
- Distance from normalized features
  - To compute distance, Do dot product
  - We can use convolution operation

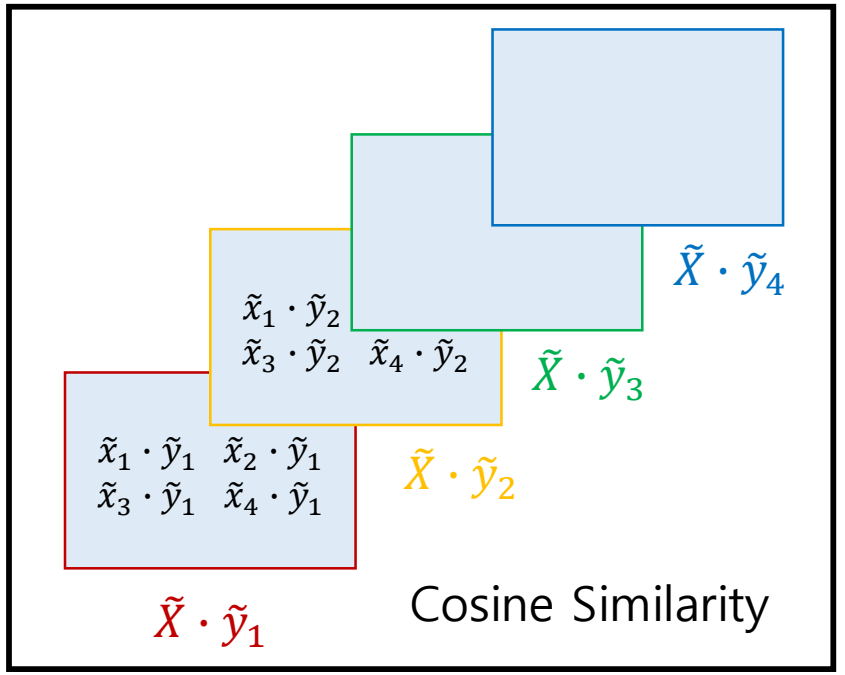




$$d_{ij} = (1 - \frac{(x_i - \mu_y)\cdot(y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N}\sum_j y_j.$$

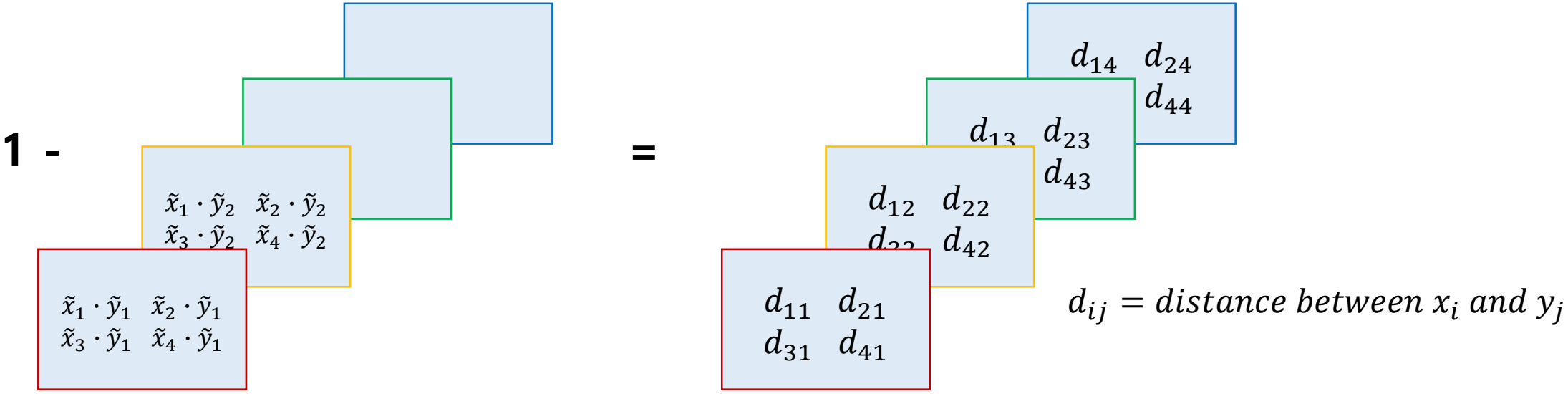
- Distance from normalized features
  - To compute distance, Do dot product
  - We can use convolution operation





$$d_{ij} = (1 - \frac{(x_i - \mu_y)\cdot(y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N}\sum_j y_j.$$

- Distance from normalized features

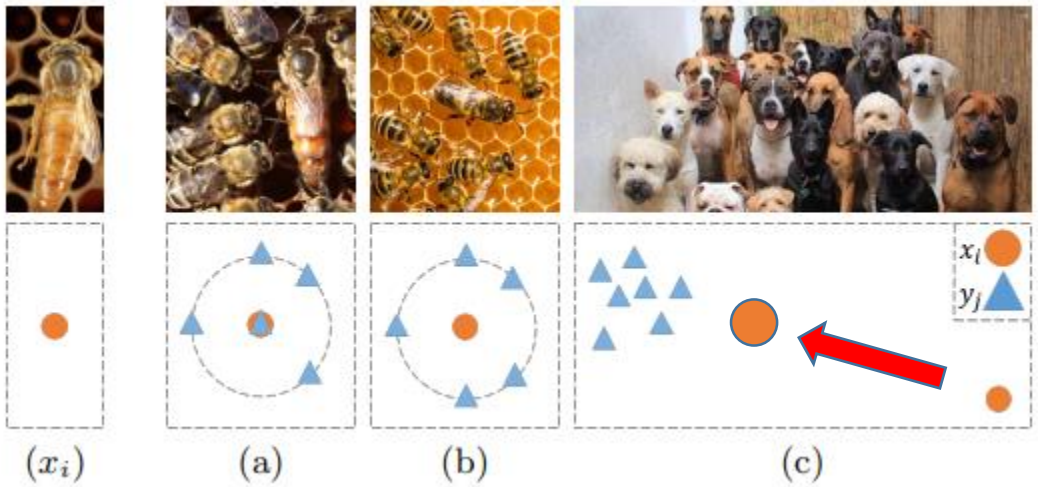


Cosine Similarity

Cosine Distance

$$d_{ij} = (1 - \frac{(x_i - \mu_y) \cdot (y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N} \sum_j y_j.$$

- x_i is not closer to any particular y_j , then its contextual similarity to all y_j should be low. This approach is robust to the scale of the distances. we start by normalizing the distances.



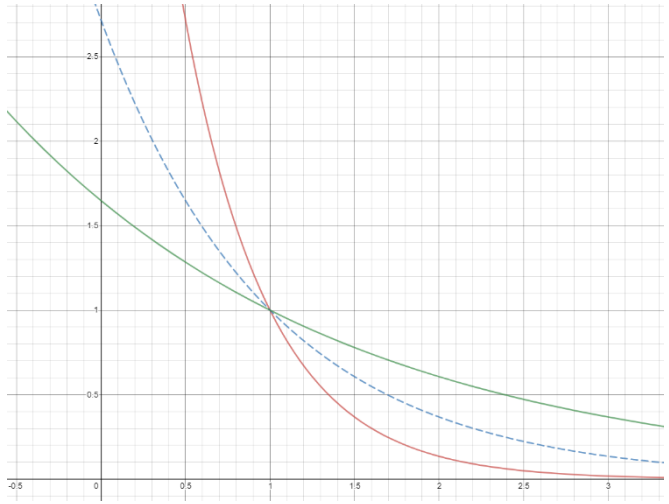
$$d_{ij} = (1 - \frac{(x_i - \mu_y)\cdot(y_j - \mu_y)}{||x_i - \mu_y||_2 ||y_j - \mu_y||_2}) \text{ where } \mu_y = \frac{1}{N}\sum_j y_j.$$

$$\tilde{d}_{ij} = \frac{d_{ij}}{\min_k d_{ik} + \epsilon} \qquad (2)$$

- We shift from distances to similarities by exponentiation where h>0 is a band-width parameter.

$$w_{ij} = \exp\left(\frac{1-\tilde{d}_{ij}}{h}\right) \tag{3}$$

Similarity



Distance

1. $y = \exp\left(\frac{(1-x)}{0.5}\right)$
2. $y = \exp\left(\frac{(1-x)}{1.0}\right)$
3. $y = \exp\left(\frac{(1-x)}{2.0}\right)$

- Finally, we define the contextual similarity between features to be a scale invariant version of the normalized similarities.

$$\mathrm{CX}_{ij} = w_{ij} / \sum_k w_{ik}$$ ⟵ *similarity between* x_i *and* y_j

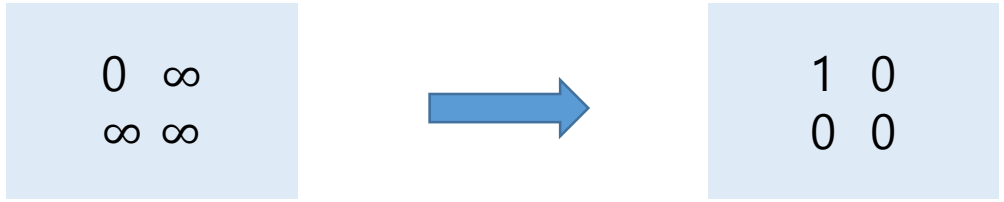
- To calculate the similarity between the images we find for each feature y_j the feature x_i that is most similar to it, and then sum the corresponding feature similarity values over all y_j.

$$\mathrm{CX}(x, y) = \mathrm{CX}(X, Y) = \frac{1}{N} \sum_j \max_i \mathrm{CX}_{ij}$$ ⟵ *similarity between* X *and* Y

- The contextual loss is defined as: $\mathcal{L}_{\mathrm{CX}}(x, y, l) = -\log\left(\mathrm{CX}\left(\Phi^l(x), \Phi^l(y)\right)\right)$

- ***Extreme cases***

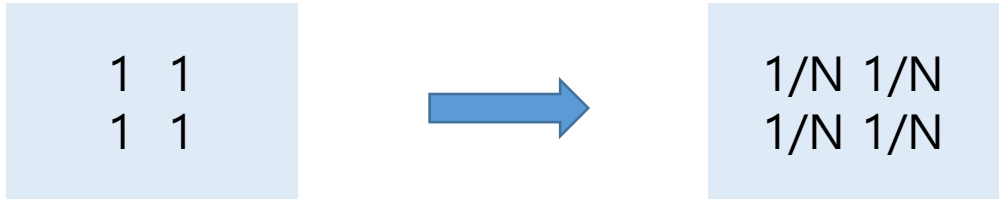
- Since the Contextual Similarity sums over normalized values we get that $CX(X,Y) \in [0,1]$. Comparing an image to itself yields $CX(X,X) = 1$, since the feature similarity values will be $CX_{ij} = 1$ and 0 otherwise.



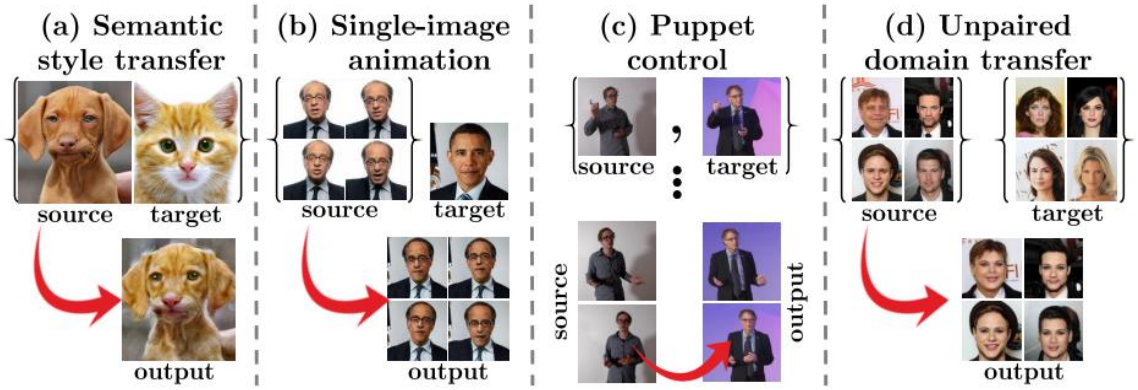
$\tilde{d}_{ij}$ CX_{ij}

- ***Extreme cases***

- At the other extreme, when the sets of features are far from each other then $CX_{ij} \approx \frac{1}{N} \forall i, j$, and thus $\mathrm{CX}(\mathrm{X}, \mathrm{Y}) \approx \frac{1}{N} \to 0$.



Applications



Application	Architecture	Loss function: Proposed	Loss function: Previous	Paired	Aligned
Style transfer	Optim. [6]	$\mathcal{L}_{\text{CX}}^{t}+\mathcal{L}_{\text{CX}}^{s}$	$\mathcal{L}_{Gram}^{t}+\mathcal{L}_{P}^{s}$	✓	✗
Single-image animation	CRN [10]	$\mathcal{L}_{\text{CX}}^{t}+\mathcal{L}_{\text{CX}}^{s}$	$\mathcal{L}_{Gram}^{t}+\mathcal{L}_{P}^{s}$	✓	✗
Puppet control	CRN [10]	$\mathcal{L}_{\text{CX}}^{t}+\mathcal{L}_{P}^{t}$	$\mathcal{L}_{1}^{t}+\mathcal{L}_{P}^{t}$	✓	✓✗
Domain transfer	CRN [10]	$\mathcal{L}_{\text{CX}}^{t}+\mathcal{L}_{\text{CX}}^{s}$	CycleGAN[2]	✗	✗

Table 1. Applications settings: A summary of the settings for our four applications.

- $L^t = L(G(s), t)$ for similarity between the generated image $G(s)$ and the target t.
- $L^s = L(G(s), s)$ to demand similarity to the source image s.

- Semantic Style Transfer

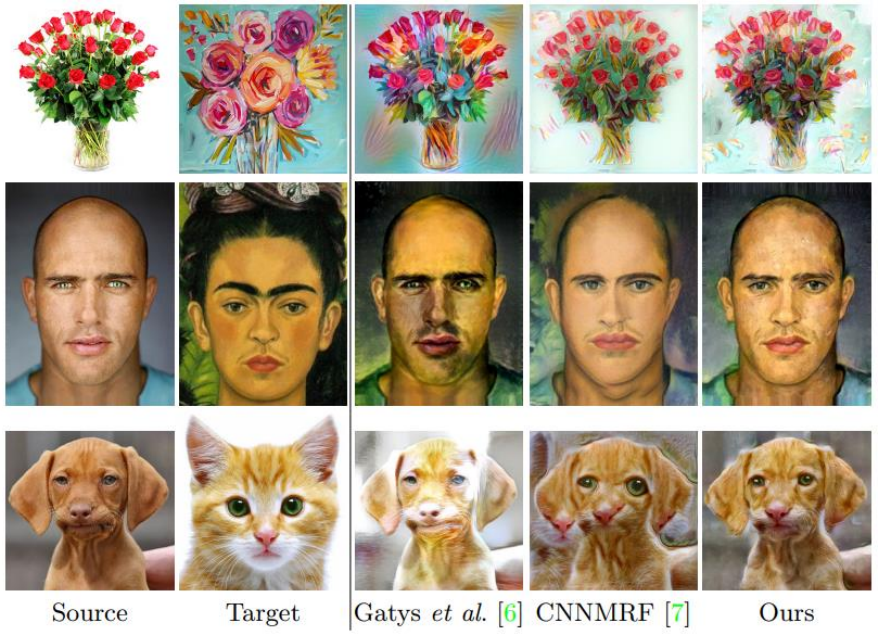


Fig. 7. Semantic style transfer: The Contextual loss naturally provides semantic style transfer across regions of corresponding semantic meaning. Notice how in our results: (row1) the flowers and the stalks changed their style correctly, (row2) the man's eyebrows got connected, a little mustache showed up and his lips changed their shape and color, and (row3) the cute dog got the green eyes, white snout and yellowish head of the target cat. Our results are much different from those of [6] that transfer the style globally over the entire image. CNNMRF [7] achieves semantic matching but is very prone to artifacts. See supplementary for many more results and comparisons.

- Single Image Animation & Puppet control
- The Contextual Loss for Image Transformation with Non-Aligned Data – YouTube
- 8:26~
- 9:50~

- Unpaired domain transfer

Fig. 11. Unpaired domain transfer: Gender transformation with unpaired data (CelebA) [36], (Top) Male-to-female, (Bottom) Female-to-male. Our approach successfully modifies the facial attributes making the men more feminine (or the women more masculine) while preserving the original person identity. The changes are mostly noticeable in the eye makeup, eyebrows shaping and lips. Our gender modification is more successful than that of CycleGAN [2], even though we use a single feed-forward network, while they train a complex 4-network architecture.