The Contextual Loss for Image Transformation with Non-Aligned Data

Roey Mechrez* , Itamar Talmi*, Lihi Zelnik-Manor

Technion - Israel Institute of Technology
{roey@campus,titamar@campus,lihi@ee}.technion.ac.il

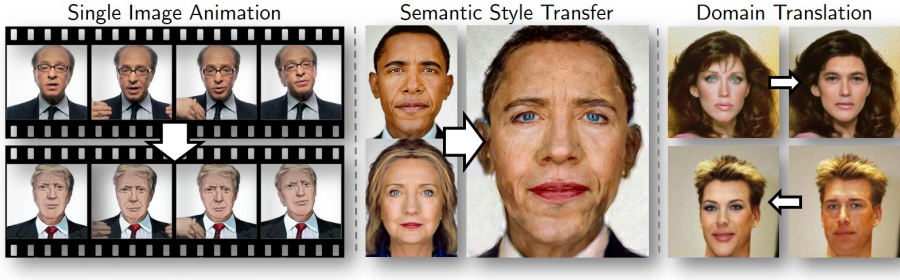


Fig. 1. Our Contextual loss is effective for many image transformation tasks: It can make a Trump cartoon imitate Ray Kurzweil, give Obama some of Hillary's features, and, turn women more masculine or men more feminine. Mutual to these tasks is the absence of ground-truth targets that can be compared pixel-to-pixel to the generated images. The Contextual loss provides a simple solution to all of these tasks.

Abstract. Feed-forward CNNs trained for image transformation problems rely on loss functions that measure the similarity between the generated image and a target image. Most of the common loss functions assume that these images are spatially aligned and compare pixels at corresponding locations. However, for many tasks, aligned training pairs of images will not be available. We present an alternative loss function that does not require alignment, thus providing an effective and simple solution for a new space of problems. Our loss is based on both context and semantics – it compares regions with similar semantic meaning, while considering the context of the entire image. Hence, for example, when transferring the style of one face to another, it will translate eyes-to-eyes and mouth-to-mouth. Our code can be found at https://www.github.com/roimehrez/contextualLoss

1 Introduction

Many classic problems can be framed as image transformation tasks, where a system receives some source image and generates a corresponding output image.

* indicate authors contributed equally

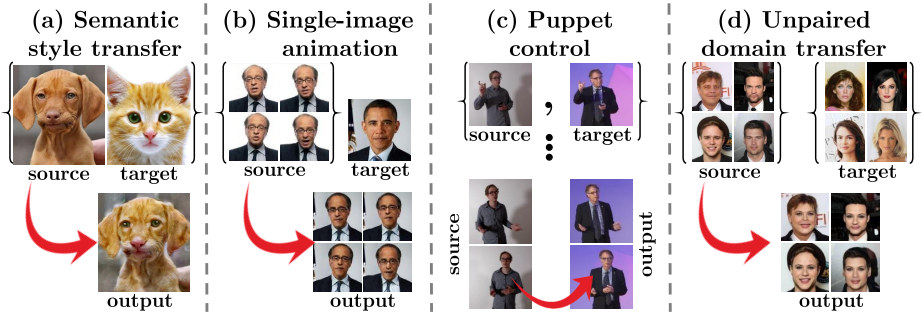


Fig. 2. Non-aligned data: In many image translation tasks the desired *output* images are *not* spatially aligned with any of the available *target* images. (a) In *semantic style transfer* regions in the output image should share the style of corresponding regions in the target, e.g., the dog's fur, eyes and nose should be styled like those of the cat. (b) In *single-image animation* we animate a single target image according to input animation images. (c) In *puppet control* we animate a target "puppet" according to an input "driver" but we have available multiple training pairs of driver-puppet images. (d) In *domain transfer*, e.g, gender translation, the training images are not even paired, hence, clearly the outputs and targets are not aligned.

Examples include image-to-image translation [1,2], super-resolution [3,4,5], and style-transfer [6,7,8]. Samples of our results for some of these applications are presented in Figure 1.

One approach for solving image transformation tasks is to train a feed-forward convolutional neural network. The training is based on comparing the image generated by the network with a target image via a differentiable loss function. 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., $L2$ [3,9], $L1$ [1,2,10], and the perceptual loss of [8] (often computed at a coarse level). (ii) Global loss functions, such as the Gram loss [6], which successfully captures style [6,8] and texture [4,11] by comparing statistics collected over the entire image. Orthogonal to these are adversarial loss functions (GAN) [12], that push the generated image to be of high likelihood given examples from the target domain. This is complementary and does not compare the generated and the target image directly.

Both types of image comparison loss functions have been shown to be highly effective for many tasks, however, there are some cases they do not address. Specifically, the pixel-to-pixel loss functions explicitly assume that the generated image and target image are spatially aligned. They are not designed for problems where the training data is, by definition, not aligned. This is the case, as illustrated in Figures 1 & 2, in tasks such as semantic style transfer, single-image animation, puppet control, and unpaired domain translation. Non-aligned images can be compared by the Gram loss, however, due to its global nature it

translates global characteristics to the entire image. It cannot be used to constrain the content of the generated image, which is required in these applications.

In this paper we propose the *Contextual Loss* – a loss function targeted at non-aligned data. Our key idea is to treat an image as a collection of features, and measure the similarity between images, based on the similarity between their features, ignoring the spatial positions of the features. We form matches between features by considering all the features in the generated image, thus incorporating global image context into our similarity measure. Similarity between images is then defined based on the similarity between the matched features. This approach allows the generated image to spatially deform with respect to the target, which is the key to our ability to solve all the applications in Figure 2 with a feed-forward architecture. In addition, the Contextual loss is not overly global (which is the main limitation of the Gram loss) since it compares features, and therefore regions, based on semantics. This is why in Figure 1 style-transfer endowed Obama with Hillary's eyes and mouth, and domain translation changed people's gender by shaping/thickening their eyebrows and adding/removing makeup.

A nice characteristic of the Contextual loss is its tendency to maintain the appearance of the target image. This enables generation of images that look real even without using GANs, whose goal is specifically to distinguish between 'real' and 'fake', and are sometimes difficult to fine tune in training.

We show the utility and benefits of the Contextual loss through the applications presented in Figure 2. In all four applications we show state-of-the-art or comparable results without using GANs. In style transfer, we offer an advancement by translating style in a semantic manner, without requiring segmentation. In the tasks of puppet-control and single-image-animation we show a significant improvement over previous attempts, based on pixel-to-pixel loss functions. Finally, we succeed in domain translation without paired data, outperforming CycleGAN [2], even though we use a single feed-forward network, while they train four networks (two generators, and two discriminators).

2 Related Work

Our key contribution is a new loss function that could be effective for many image transformation tasks. We review here the most relevant approaches for solving image-to-image translation and style transfer, which are the applications domains we experiment with.

Image-to-Image Translation includes tasks whose goal is to transform images from an input domain to a target domain, for example, day-to-night, horse-to-zebra, label-to-image, BW-to-color, edges-to-photo, summer-to-winter, photo-to-painting and many more. Isola *et al.* [1] (pix2pix) obtained impressive results with a feed-forward network and adversarial training (GAN) [12]. Their solution demanded pairs of aligned input-target images for training the network with a pixel-to-pixel loss function ($L2$ or $L1$). Chen and Koltun [10] proposed a Cascaded Refinement Network (CRN) for solving label-to-image, where an image is

generated from an input semantic label map. Their solution as well used pixel-to-pixel losses, (Perceptual [8] and $L1$), and was later appended with GAN [13]. These approaches require paired and aligned training images.

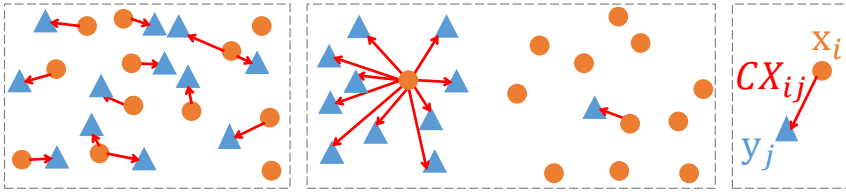
Domain transfer has recently been applied also for problems were paired training data is not available [2,14,15]. To overcome the lack of training pairs the simple feed-forward architectures were replaced with more complex ones. The key idea being that translating from one domain to the other, and then going back, should take us to our starting point. This was modeled by complex architectures, e.g., in CycleGAN [2] four different networks are required. The circular process sometimes suffers from the mode collapse problem, a prevalent phenomenon in GANs, where data from multiple modes of a domain map to a single mode of a different domain [14].

Style Transfer aims at transferring the style of a target image to an input image [16,17,18,19]. Most relevant to our study are approaches based on CNNs. These differ mostly in the choice of architecture and loss function [6,7,8,20,21]. Gatys *et al.* [6] presented stunning results obtained by optimizing with a gradient based solver. They used the pixel-to-pixel Perceptual loss [8] to maintain similarity to the input image and proposed the Gram loss to capture the style of the target. Their approach allows for arbitrary style images, but this comes at a high computational cost. Methods with lower computational cost have also been proposed [8,21,22,23]. The speedup was obtained by replacing the optimization with training a feed-forward network. The main drawback of these latter methods is that they need to be re-trained for each new target style.

Another line of works aim at *semantic* style transfer, were the goal is to transfer style across regions of corresponding semantic meaning, e.g., sky-to-sky and trees-to-trees (in the methods listed above the target style is transfered globally to the entire image). One approach is to replace deep features of the input image with matching features of the target and then invert the features via efficient optimization [20] or through a pre-trained decoder [24]. Li *et al.* [7] integrate a Markov Random Field into the output synthesis process (CNNMRF). Since the matching in these approaches is between neural features semantic correspondence is obtained. A different approach to semantic style transfer is based on segmenting the image into regions according to semantic meaning [25,26]. This leads to semantic transfer, but depends on the success of the segmentation process. In [27] a histogram loss was suggested in order to synthesize textures that match the target statistically. This improves the color fatefulness but does not contribute to the semantic matching. Finally, there are also approaches tailored to a specific domain and style, such as faces or time-of-day in city-scape images [28,29].

3 Method

Our goal is to design a loss function that can measure the similarity between images that are not necessarily aligned. Comparison of non-aligned images is also



(a) Similar (b) Not-similar

Fig. 3. Contextual Similarity between images: Orange circles represent the features of an image x while the blue triangles represent the features of a target image y. The red arrows match each feature in y with its most *contextually similar* (Eq.(4)) feature in x. (a) Images x and y are similar: many features in x are matched with similar features in y. (b) Images x and y are not-similar: many features in x are not matched with any feature in y. The Contextual loss can be thought of as a weighted sum over the red arrows. It considers only the features and not their spatial location in the image.

the core of template matching methods, that look for image-windows that are similar to a given template under occlusions and deformations. Recently, Talmi *et al.* [30] proposed a statistical approach for template matching with impressive results. Their measure of similarity, however, has no meaningful derivative, hence, we cannot adopt it as a loss function for training networks. We do, nonetheless, draw inspiration from their underlying observations.

3.1 Contextual Similarity between Images

We start by defining a measure of similarity between a pair of images. Our key idea is to represent each image as a set of high-dimensional points (features), and consider two images as similar if their corresponding sets of points are similar. As illustrated in Figure 3, we consider a pair of images as similar when for most features of one image there exist similar features in the other. Conversely, when the images are different from each other, many features of each image would have no similar feature in the other image. Based on this observation we formulate the contextual similarity measure between images.

Given an image x and a target image y we represent each as a collection of points (e.g., VGG19 features [31]): $X = \{x_i\}$ and $Y = \{y_j\}$. We assume $|Y| = |X| = N$ (and sample N points from the bigger set when $|Y| \neq |X|$). 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. Formally, the contextual similarity between images is defined as:

$$\mathrm{CX}(x, y) = \mathrm{CX}(X, Y) = \frac{1}{N} \sum_j \max_i \mathrm{CX}_{ij} \tag{1}$$

where CX_{ij}, to be defined next, is the similarity between features x_i and y_j.

We incorporate global image context via our definition of the similarity CX_{ij} between features. Specifically, we consider feature x_i as contextually similar to

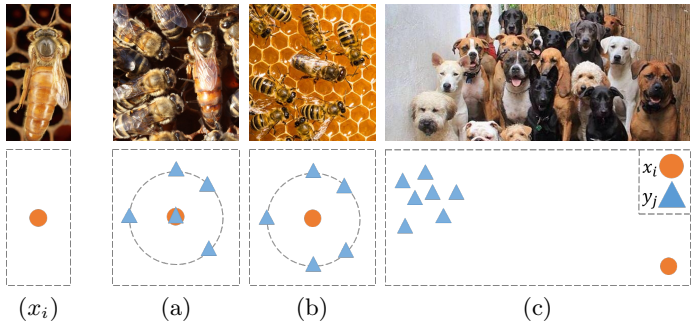


Fig. 4. Contextual similarity between features: We define the contextual similarity CX_{ij} between features x_i (queen bee) and y_j by considering the context of all the features in y. (a) x_i overlaps with a single y_j (the queen bee) while being far from all others (worker bees), hence, its contextual similarity to it is high while being low to all others. (b) x_i is far from all y_j's (worker bees), hence, its contextual similarity to all of them is low. (c) x_i is very far (different) from all y_j's (dogs), however, for scale robustness the contextual similarity values here should resemble those in (b).

feature y_j if it is significantly closer to it than to all other features in Y. When this is not the case, i.e., 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, e.g., if x_i is far from all y_j then CX_{ij} will be low $\forall j$ regardless of how far apart x_i is. Figure 4 illustrates these ideas via examples.

We next formulate this mathematically. Let d_{ij} be the Cosine distance between x_i and y_j[1]. We consider features x_i and y_j as similar when $d_{ij} \ll d_{ik}, \forall k \neq j$. To capture this we start by normalizing the distances:

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

for a fixed $\epsilon = 1\mathrm{e}{-5}$. We shift from distances to similarities by exponentiation:

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

where $h > 0$ is a band-width parameter. Finally, we define the contextual similarity between features to be a scale invariant version of the normalized similarities:

$$\mathrm{CX}_{ij} = w_{ij} / \sum_k w_{ik} \tag{4}$$

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

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

extreme, when the sets of features are far from each other then $\mathrm{CX}_{ij} \approx \frac{1}{N}\,\forall i,j$, and thus $\mathrm{CX}(X,Y) \approx \frac{1}{N} \to 0$. We further observe that binarizing the values by setting $\mathrm{CX}_{ij} = 1$ if $w_{ij} > w_{ik}, \forall k \neq j$ and 0 otherwise, is equivalent to finding the Nearest Neighbor in Y for every feature in X. In this case we get that $\mathrm{CX}(X,Y)$ is equivalent to counting how many features in Y are a Nearest Neighbor of a feature in X, which is exactly the template matching measure proposed by [30].

3.2 The Contextual loss

For training a generator network we need to define a loss function, based on the contextual similarity of Eq.(1). Let x and y be two images to be compared. We extract the corresponding set of features from the images by passing them through a perceptual network Φ, where in all of our experiments Φ is VGG19 [31]. Let $\Phi^l(x)$, $\Phi^l(y)$ denote the feature maps extracted from layer l of the perceptual network Φ of the images x and y, respectively. The contextual loss is defined as:

$$\boxed{\mathcal{L}_{\mathrm{CX}}(x,y,l) = -\log\left(\mathrm{CX}\left(\Phi^l(x), \Phi^l(y)\right)\right)} \tag{5}$$

In image transformation tasks we train a network G to map a given source image s into an output image $G(s)$. To demand similarity between the generated image and the target we use the loss $\mathcal{L}_{\mathrm{CX}}(G(s),t,l)$. Often we demand also similarity to the source image by the loss $\mathcal{L}_{\mathrm{CX}}(G(s),s,l)$. In Section 4 we describe in detail how we use such loss functions for various different applications and what values we select for l.

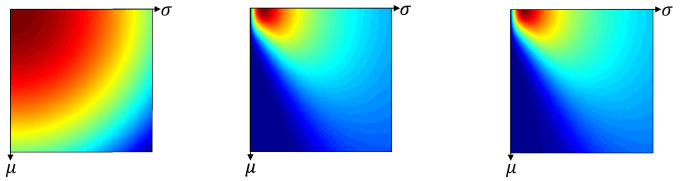
Other loss functions: In the following we compare the Contextual loss to other popular loss functions. We provide here their definitions for completeness:

- 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].

The first two are pixel-to-pixel loss functions that require alignment between the images x and y. The Gram loss is global and robust to pixel locations.

3.3 Analysis of the Contextual Loss

Expectation Analysis: The Contextual loss compares sets of features, thus implicitly, it can be thought of as a way for comparing distributions. To support this observation we provide empirical statistical analysis, similar to that presented in [30,32]. Our goal is to show that the expectation of $\mathrm{CX}(X,Y)$ is maximal when the points in X and Y are drawn from the same distribution, and drops sharply as the distance between the two distributions increases. This is done via a simplified mathematical model, in which each image is modeled as



(a) $E[\mathbf{L2}]$ (b) $E[\mathbf{DIS}]$ (c) $E[\mathbf{CX}]\,(h=0.1)$

Fig. 5. Expected behavior in the 1D Gaussian case: Two point sets, X and Y, are generated by sampling $N=M=100$ points from $N(0;1)$, and $N(\mu;\sigma)$, respectively, with $[\mu,\sigma]\in[0,10]$. The approximated expectations of (a) $L2$ (from [32]), (b) DIS (from [30]), and, (c) the proposed CX, as a function of μ and σ show that CX drops much more rapidly than $L2$ as the distributions move apart.

a set of points drawn from a 1D Gaussian distribution. We compute the similarity between images for varying distances between the underlying Gaussians. Figure 5 presents the resulting approximated expected values. It can be seen that $\mathrm{CX}(X,Y)$ is likely to be maximized when the distributions are the same, and falls rapidly as the distributions move apart from each other. Finally, similar to [30,32], one can show that this holds also for the multi-dimensional case.

Toy experiment with non-aligned data: In order to examine the robustness of the contextual loss to non-aligned data, we designed the following toy experiment. Given a single noisy image s, and multiple clean images of the same scene (targets t^k), the goal is to reconstruct a clean image $G(s)$. The target images t^k are not aligned with the noisy source image s. In our toy experiment the source and target images were obtained by random crops of the same image, with random translations $\in[-10,10]$ pixels. We added random noise to the crop selected as source s. Reconstruction was performed by iterative optimization using gradient descent where we directly update the image values of s. That is, we minimize the objective function $\mathcal{L}(s,t^k)$, where $\mathcal{L}$ is either $\mathcal{L}_{\mathrm{CX}}$ or $\mathcal{L}_1$, and we iterate over the targets t^k. In this specific experiment the features we use for the contextual loss are vectorized RGB patches of size 5×5 with stride 2 (and not VGG19).

The results, presented in Figure 6, show that optimizing with $\mathcal{L}_1$ yields a drastically blurred image, because it cannot properly compare non-aligned images. The contextual loss, on the other hand, is designed to be robust to spatial deformations. Therefore, optimizing with $\mathcal{L}_{\mathrm{CX}}$ leads to complete noise removal, without ruining the image details.

We refer to reader to [33], were additional theoretical and empirical analysis of the contextual loss is presented.

4 Applications

We experiment on the tasks presented in Figure 2. To asses the contribution of the proposed loss function we adopt for each task a state-of-the-art architecture

(a) Noisy input (b) Clean targets (c) $\mathcal{L}_1$ as loss (d) $\mathcal{L}_{\text{CX}}$ 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 $\mathcal{L}_{\text{CX}}$ removes the noise nicely (d). This is since $\mathcal{L}_{\text{CX}}$ is robust to misalignments and spatial deformations.

		Loss function			
Application	**Architecture**	**Proposed**	**Previous**	**Paired**	**Aligned**
Style transfer	Optim. [6]	$\mathcal{L}^t_{\text{CX}}+\mathcal{L}^s_{\text{CX}}$	$\mathcal{L}^t_{Gram}+\mathcal{L}^s_P$	✓	✗
Single-image animation	CRN [10]	$\mathcal{L}^t_{\text{CX}}+\mathcal{L}^s_{\text{CX}}$	$\mathcal{L}^t_{Gram}+\mathcal{L}^s_P$	✓	✗
Puppet control	CRN [10]	$\mathcal{L}^t_{\text{CX}}+\mathcal{L}^t_P$	$\mathcal{L}^t_1+\mathcal{L}^t_P$	✓	✓✗
Domain transfer	CRN [10]	$\mathcal{L}^t_{\text{CX}}+\mathcal{L}^s_{\text{CX}}$	CycleGAN[2]	✗	✗

Table 1. Applications settings: A summary of the settings for our four applications. We use here simplified notations: $\mathcal{L}^t$ marks which loss is used between the generated image $G(s)$ and the target t. Similarly, $\mathcal{L}^s$ stands for the loss between $G(s)$ and the source (input) s. We distinguish between paired and unpaired data and between semi-aligned (x+v) and non-aligned data. Definitions of the loss functions are in the text.

and modify only the loss functions. In some tasks we also compare to other recent solutions. For all applications we used TensorFlow [34] and Adam optimizer [35] with the default parameters ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-08$). Unless otherwise mentioned we set $h=0.5$ (of Eq. (3)).

The tasks and the corresponding setups are summarized in Table 1. We use shorthand notation $\mathcal{L}^t_{type} = \mathcal{L}_{type}(G(s), t, l)$ to demand similarity between the generated image $G(s)$ and the target t and $\mathcal{L}^s_{type} = \mathcal{L}_{type}(G(s), s, l)$ to demand similarity to the source image s. The subscripted notation $\mathcal{L}_{type}$ stands for either the proposed $\mathcal{L}_{\text{CX}}$ or one of the common loss functions defined in Section 3.2.

4.1 Semantic Style Transfer

In style-transfer the goal is to translate the style of a target image t onto a source image s. A landmark approach, introduced by Gatys *et al.* [6], is to minimize a combination of two loss functions, the perceptual loss $\mathcal{L}_P(G(s), s, l_P)$ to maintain the content of the source image s, and the Gram loss $\mathcal{L}_{Gram}(G(s), t, l_{\mathcal{G}})$ to enforce style similarity to the target t (with $l_{\mathcal{G}} = \{conv\mathbf{k}\_1\}^5_{\mathbf{k}=1}$ and $l_P = conv4\_2$).

We claim that the Contextual loss is a good alternative for both. By construction it makes a good choice for the style term, as it does not require alignment.

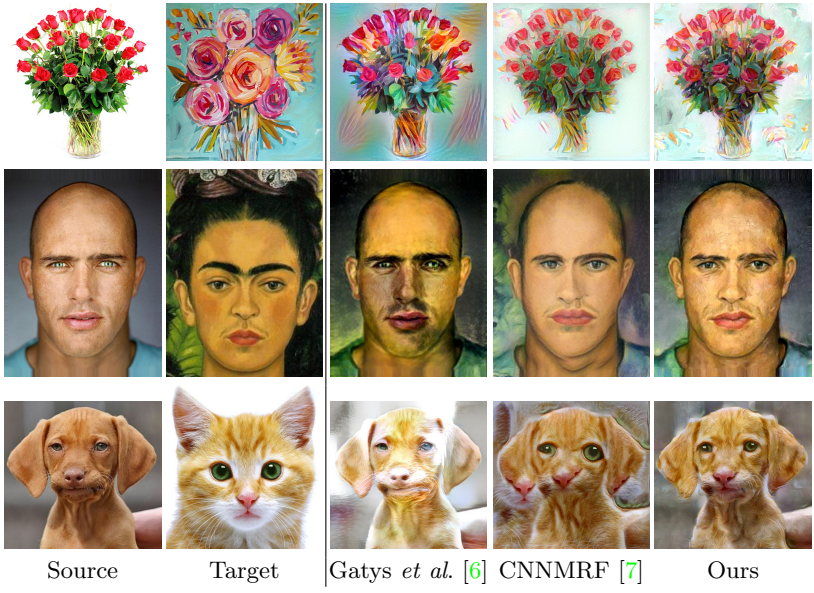


Fig. 7. Semantic style transfer: The Contextual loss naturally provides semantic style transfer across regions of corresponding semantic meaning. Notice how in our results: (row1) the flowers and the stalks changed their style correctly, (row2) the man's eyebrows got connected, a little mustache showed 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.

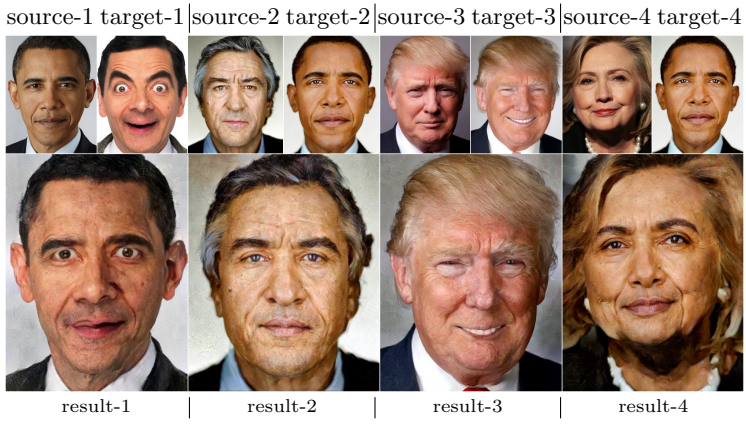


Fig. 8. Playing with target: Results of transferring different target targets. Notice how in each result we mapped features semantically, transferring shapes, colors and textures to the hair, mouth, nose, eyes and eyebrows. It is nice to see how Trump got a smile full of teeth and Hilary was marked with Obama's mole.

Moreover, it will allow transferring style features between regions according to their semantic similarity, rather than globally over the entire image, which is what one gets with the Gram loss. The Contextual loss is also a good choice for the content term since it demands similarity to the source, but allows some positional deformations. Such deformations are advantageous, since due to the style change the stylized and source images will not be perfectly aligned.

To support these claims we adopt the optimization-based framework of Gatys *et al.* [6][2], that directly minimizes the loss through an iterative process, and replace their objective with:

$$\mathcal{L}(G) = \mathcal{L}_{\mathrm{CX}}(G(s), t, l_t) + \mathcal{L}_{\mathrm{CX}}(G(s), s, l_s) \tag{6}$$

where $l_s = conv4\_2$ (to capture content) and $l_t = \{conv\mathbf{k}\_2\}_{\mathbf{k}=2}^{4}$ (to capture style). We set h as 0.1 and 0.2 for the content term and style term respectively. In our implementation we reduced memory consumption by random sampling of layer $conv2\_2$ into 65×65 features.

Figure 8 presents a few example results. It can be seen that the style is transfered across corresponding regions, e.g., eyes-to-eyes, hair-to-hair, etc. In Figure 7 we compare our style transfer results with two other methods: Gatys *et al.* [6] and CNNMRF [7]. The only difference between our setup and theirs is the loss function, as all three use the same optimization framework. It can be seen that our approach transfers the style semantically across regions, whereas, in Gatys' approach the style is spread all over the image, without semantics. CNN-MRF, on the other hand, does aim for semantic transfer. It is based on nearest neighbor matching of features, which indeed succeeds in replacing semantically corresponding features, however, it suffers from severe artifacts.

4.2 Single Image Animation

In single-image animation the data consists of many animation images from a source domain (e.g, person $\mathcal{S}$) and only a single image t from a target domain (e.g., person $\mathcal{T}$). The goal is to animate the target image according to the input source images. This implies that by the problem definition the generated images $G(s)$ are not aligned with the target t.

This problem setup is naturally handled by the Contextual loss. We use it both to maintain the animation (spatial layout) of the source s and to maintain the appearance of the target t:

$$\mathcal{L}(G) = \mathcal{L}_{\mathrm{CX}}(G(s), t, l_t) + \mathcal{L}_{\mathrm{CX}}(G(s), s, l_s) \tag{7}$$

where $l_s = conv4\_2$ and $l_t = \{conv3\_2, conv4\_2\}$. We selected the CRN architecture of [10][3] and trained it for 10 epochs on 1000 input frames.

Results are shown in Figure 9. We are not aware of previous work the solves this task with a generator network. We note, however, that our setup is somewhat related to fast style transfer [8], since effectively the network is trained to

[2] We used the implementation in https://github.com/anishathalye/neural-style

[3] We used the original implementation http://cqf.io/ImageSynthesis/

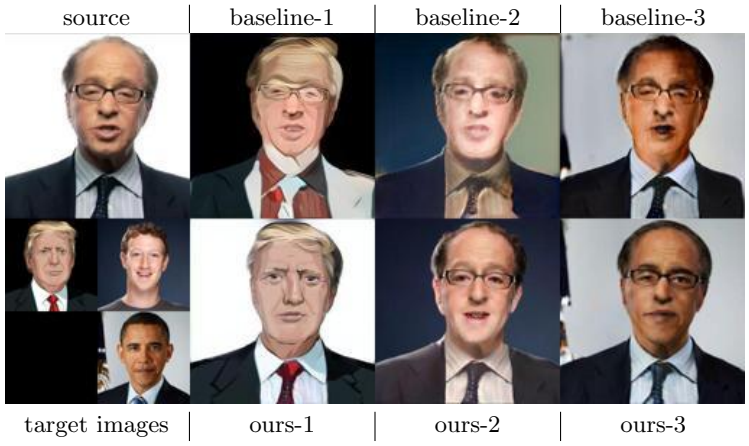


Fig. 9. Single Image Animation: This figure is an animated gif showing every 20th frame from the test-set (animation works only in Acrobat Reader, video provided in supplementary). Given an input video (top-left) we animate three different target images (bottom-left). Comparing our animations (bottom) with the baseline (top) shows that we are much more faithful to the appearance of the targets and the motions of the input. Note, that our solution and the baseline differ only in the loss functions.

generate images with content similar to the input (source) but with style similar to the target. Hence, as baseline for comparison, we trained the same CRN architecture and replaced only the objective with a combination of the Perceptual (with $l_P = conv5\_2$) and Gram losses (with $l_\mathcal{G} = \{conv\mathbf{k}\_1\}_{\mathbf{k}=1}^{5}$), as proposed by [8]. It can be seen that using our Contextual loss is much more successful, leading to significantly fewer artifacts.

4.3 Puppet control

Our task here is somewhat similar to single-image animation. We wish to animate a target "puppet" according to provided images of a "driver" person (the source). This time, however, available to use are training pairs of source-target (driver-puppet) images, that are semi-aligned. Specifically, we repeated an experiment published online, were Brannon Dorsey (the driver) tried to control Ray Kurzweil (the puppet)[4]. For training he filmed a video ($\sim$1K frames) of himself imitating Kurzweil's motions. Then, given a new video of Brannon, the goal is to generate a corresponding animation of the puppet Kurzweil.

The generated images should look like the target puppet, hence we use the Contextual loss to compare them. In addition, since in this particular case the training data available to us consists of pairs of images that are semi-aligned, they do share a very coarse level similarity in their spatial arrangement. Hence, to further refine the optimization we add a Perceptual loss, computed at a very

[4] B. Dorsey, https://twitter.com/brannondorsey/status/808461108881268736

Fig. 10. Puppet control: Results of animating a "puppet" (Ray Kurzweil) according to the input video shown on the left. Our result is sharper, less prone to artifacts and more faithful to the input pose and the "puppet" appearance. This figure is an animated gif showing every 10th frame from the test-set (animation seen only in Acrobat Reader, video provided in the project page).

coarse level, that does not require alignment. Our overall objective is:

$$\mathcal{L}(G) = \mathcal{L}_{\text{CX}}(G(s), t, l_{\text{CX}}) + \lambda_P \cdot \mathcal{L}_P(G(s), t, l_P) \tag{8}$$

where $l_{\text{CX}} = \{conv\mathbf{k}\_2\}_{\mathbf{k}=2}^{4}$, $l_P = conv5\_2$, and $\lambda_P = 0.1$ to let the contextual loss dominate. As architecture we again selected CRN [10] and trained it for 20 epochs.

We compare our approach with three alternatives: (i) Using the exact same CRN architecture, but with the pixel-to-pixel loss function $\mathcal{L}_1$ instead of $\mathcal{L}_{\text{CX}}$. (ii) The Pix2pix architecture of [1] that uses $\mathcal{L}_1$ and adversarial training (GAN), since this was the original experiment. (iii) We also compare to CycleGAN [2] that treats the data as unpaired and compares images with $\mathcal{L}_1$ and uses adversarial training (GAN). Results are presented in Figure 10. It can be seen that the puppet animation generated with our approach is much sharper, with significantly fewer artifacts, and captures nicely the poses of the driver, even though we don't use GAN.

4.4 Unpaired domain transfer

Finally, we use the Contextual loss also in the unpaired scenario of domain transfer. We experimented with gender change, i.e., making male portraits more feminine and vice versa. Since the data is unpaired (i.e., we do not have the female versions of the male images) we sample random pairs of images from the two domains. As the Contextual loss is robust to misalignments this is not a problem. We use the exact same architecture and loss as in single-image-animation.

Our results, presented in Figure 11, are quite successful when compared with CycleGAN [2]. This is a nice outcome since our approach provides a much simpler alternative – while the CycleGAN framework trains four networks (two generators and two discriminators), our approach uses a single feed-forward generator network (without GAN). This is possible because the Contextual loss does not require aligned data, and hence, can naturally train on non-aligned random pairs.

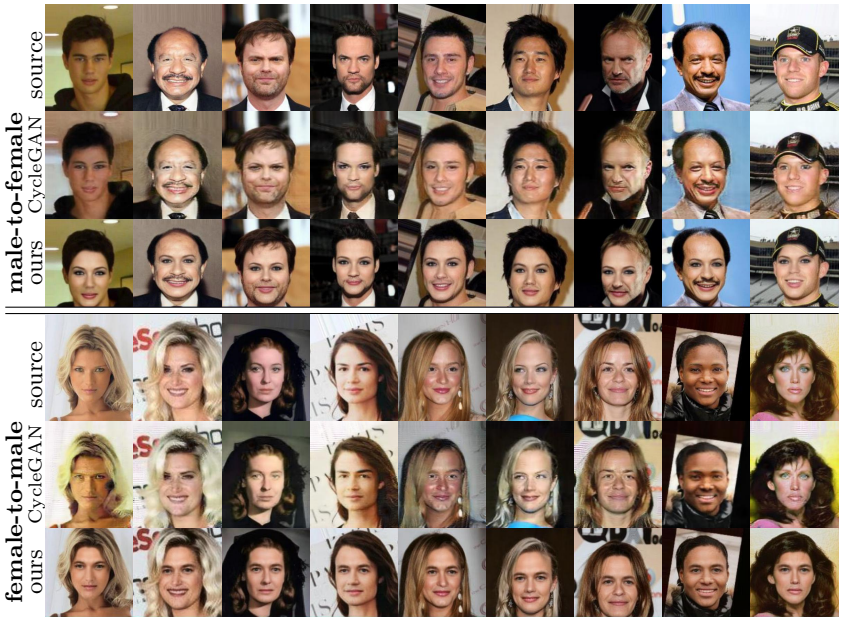


Fig. 11. Unpaired domain transfer: Gender transformation with unpaired data (CelebA) [36], (Top) Male-to-female, (Bottom) Female-to-male. Our approach successfully modifies the facial attributes making the men more feminine (or the women more masculine) while preserving the original person identity. 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.

5 Conclusions

We proposed a novel loss function for image generation that naturally handles tasks with non-aligned training data. We have applied it for four different applications and showed state-of-the-art (or comparable) results on all.

In our follow-up work, [33], we suggest to use the Contextual loss for realistic restoration, specifically for the tasks of super-resolution and surface normal estimation. We draw a theoretical connection between the Contextual loss and KL-divergence, which is supported by empirical evidence. In future work we hope to seek other loss functions, that could overcome further drawbacks of the existing ones.

In the supplementary we present limitations of our approach, ablation studies, and explore variations of the proposed loss.

Acknowledgements: This research was supported by the Israel Science Foundation under Grant 1089/16 and by the Ollendorf foundation.

References

1. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. (2017) 2, 3, 13
2. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. (2017) 2, 3, 4, 9, 13, 14
3. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. (2017) 2
4. Sajjadi, M.S., Scholkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: ICCV. (2017) 2
5. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017) 2
6. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR. (2016) 2, 4, 7, 9, 10, 11
7. Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: CVPR. (2016) 2, 4, 10, 11
8. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. (2016) 2, 4, 7, 12
9. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: NIPS. (2014) 2
10. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV. (2017) 2, 3, 9, 11, 13
11. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Diversified texture synthesis with feed-forward networks. In: CVPR. (2017) 2
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014) 2, 3
13. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. arXiv preprint arXiv:1711.11585 (2017) 4
14. Kim, T., Cha, M., Kim, H., Lee, J., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192 (2017) 4
15. Yi, Z., Zhang, H., Gong, P.T., et al.: Dualgan: Unsupervised dual learning for image-to-image translation. arXiv preprint arXiv:1704.02510 (2017) 4
16. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Computer graphics and interactive techniques, ACM (2001) 4
17. Liang, L., Liu, C., Xu, Y.Q., Guo, B., Shum, H.Y.: Real-time texture synthesis by patch-based sampling. ACM ToG (2001) 4
18. Elad, M., Milanfar, P.: Style transfer via texture synthesis. IEEE Transactions on Image Processing (2017) 4
19. Frigo, O., Sabater, N., Delon, J., Hellier, P.: Split and match: example-based adaptive patch sampling for unsupervised style transfer. In: CVPR. (2016) 4
20. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337 (2016) 4
21. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016) 4
22. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. ICLR (2017) 4

23. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: Feed-forward synthesis of textures and stylized images. In: ICML. (2016) 1349–1357 4
24. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV. (2017) 4
25. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. In: CVPR. (2017) 4
26. Zhao, H., Rosin, P.L., Lai, Y.K.: Automatic semantic style transfer using deep convolutional neural networks and soft masks. arXiv preprint arXiv:1708.09641 (2017) 4
27. Risser, E., Wilmot, P., Barnes, C.: Stable and controllable neural texture synthesis and style transfer using histogram losses. arXiv preprint arXiv:1701.08893 (2017) 4
28. Shih, Y., Paris, S., Durand, F., Freeman, W.T.: Data-driven hallucination of different times of day from a single outdoor photo. ACM ToG (2013) 4
29. Shih, Y., Paris, S., Barnes, C., Freeman, W.T., Durand, F.: Style transfer for headshot portraits. ACM ToG (2014) 4
30. Talmi, I., Mechrez, R., Zelnik-Manor, L.: Template matching with deformable diversity similarity. In: CVPR. (2017) 5, 7, 8
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 5, 7
32. Dekel, T., Oron, S., Rubinstein, M., Avidan, S., Freeman, W.T.: Best-buddies similarity for robust template matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2021–2029 7, 8
33. Mechrez, R., Talmi, I., Shama, F., Zelnik-Manor, L.: Learning to maintain natural image statistics. arXiv preprint arXiv:1803.04626 (2018) 8, 14
34. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016) 9
35. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
36. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV. (2015) 14