Cross-Domain Cascaded Deep Translation

Oren Katzir[1][0000−0002−1514−8919], Dani Lischinski[2][0000−0002−6191−0361], and Daniel Cohen-Or[1][0000−0001−6777−7445]

[1] Tel-Aviv University
[2] Hebrew University of Jerusalem

Abstract. In recent years we have witnessed tremendous progress in unpaired image-to-image translation, propelled by the emergence of DNNs and adversarial training strategies. However, most existing methods focus on transfer of *style* and *appearance*, rather than on *shape* translation. The latter task is challenging, due to its intricate non-local nature, which calls for additional supervision. We mitigate this by descending the deep layers of a pre-trained network, where the deep features contain more semantics, and applying the translation between these deep features. Our translation is performed in a cascaded, deep-to-shallow, fashion, along the deep feature hierarchy: we first translate between the deepest layers that encode the higher-level semantic content of the image, proceeding to translate the shallower layers, conditioned on the deeper ones. We further demonstrate the effectiveness of using pre-trained deep features in the context of unconditioned image generation. Our code and trained models will be made publicly available.

Keywords: Unpaired image-to-image translation, image generation

1 Introduction

In recent years, neural networks have significantly advanced generative image modeling. With the emergence of Generative Adversarial Networks (GANs) [9], image-to-image translation methods have dramatically progressed, revolutionizing applications such as inpainting [41], super resolution [34], domain adaptation [11], and more. In particular, there have been intriguing advances in the setting of unpaired image-to-image translation through the use of cycle-consistency [39, 43], as well as other approaches [3, 15, 20, 25]. However, most existing methods acknowledge the difficulty in translating *shapes* from one domain to another, as this might entail drastic geometric deformations. Consider, for example, translating between elephants and giraffes, where one would expect the neck of an elephant to be extended, while the elephant's head should shrink. The challenge is compounded by the fact that, even within the same domain, images might exhibit extreme variations in object shape and pose, partial occlusions, and contain multiple instances of the object of interest. One might even argue that this translation task is ill-posed to begin with, and at the very least, requires high-level semantics to be accounted for.

Fig. 1: Given an image from domain A (zebras), we extract its deep features using a network pre-trained for classification, specifically VGG-19 pre-trained on ImageNet, and translate them into deep features of domain B (giraffes). We first translate high level semantics (`conv_5_1`) of the zebra to those of a giraffe, as shown by the inner pair of images. Then, we use a cascade of deep-to-shallow translators, one for each deep feature layer, to translate shallower layers, i.e. `conv_4_1` and then `conv_3_1`. The images were obtained from the deep features by feature inversion networks.

Nonetheless, several image-to-image translation methods address shape deformation, aided by supervision in the form of a foreground mask [21, 28]. In contrast, GANimorph [8] and the recently proposed TransGaGa [35] show remarkable translation without requiring additional supervision for several datasets. However, these techniques excel in controlled setting only, where the images are controlled, and the foreground separation is rather simple.

In this paper, we address the problem of unpaired image-to-image translation, without requiring foreground masks, between two different domains, where the objects of interest share some semantic similarity (e.g., four-legged mammals), whose shapes and appearances may, nevertheless, be drastically different. Our key idea is to accomplish the translation task by learning to translate between deep feature maps. Rather than learning to extract the relevant high-level semantic information for the specific pair of domains at hand, we leverage deep features extracted by a network pre-trained for image classification, thereby benefiting from its large-scale fully supervised training.

Our work is motivated by the well-known observation that neurons in the deeper layers of pre-trained classification networks represent larger receptive fields in image space, and encode high-level semantic content [42]. In other words, local activation patterns in the deeper layers may encode very different shapes in size and structure. Furthermore, Aberman et al. [1] show that semantically similar regions from different domains, e.g., dog and cat, have similar activations. That is, the encoding of a cat's eye resembles that of a dog's eye more than that of its tail. These properties are attractive, since they suggest that it might be possible to learn a *semantically consistent* translation between activation patterns produced by images from different domains, and that the resulting (reconstructed) image would be able to change drastically, hopefully bypassing the common difficulties in image-to-image translation methods.

More specifically, we learn to translate between several layers of deep feature maps, extracted from two domains by a pre-trained classification network, namely VGG-19 [30]. The translation is carried out one layer at a time in a

deep-to-shallow (coarse-to-fine) *cascaded* manner. For each layer, we adversarially train a dedicated translator that acts in the feature space of that layer. The deepest layer translator effectively learns to translate between semantically similar global structures, such as body shape or head position, as demonstrated by the middle pair of images in Fig. 1. The translator of each shallower layer is conditioned on the translation result of the previous layer, and learns to add fine scale and appearance details, such as texture. At every layer, in order to visualize the generated deep features, we use a network pre-trained for inverting the deep features of VGG-19, following the method of Dosovitskiy and Brox [5]. The images shown in Fig. 1 were generated in this manner.

Our conceptual novelty may be regarded as applying transfer learning between classification and image translation, as we learn to translate high-level semantics, encoded by the deep features extracted by a pre-trained classification network. This is in contrast to existing methods [8, 35], which learn to translate the images directly. We compare our method with several state-of-the-art image translation methods. To demonstrate the effectiveness of our approach, we present results for several pairs of domains that share some high-level semantics, yet exhibit drastically different shapes and appearances. These domains are extremely challenging, as images might contain multiple instances of the subject, with cropping and occlusion, and exhibiting a variety of poses. Nevertheless, our translations are semantically consistent, typically preserving the number of instances, and reproducing their poses, partial occlusion or cropping, as shown in Fig. 5. We further demonstrate the power of our transfer learning approach by leveraging the same deep feature spaces to train an unconditioned image generation model.

2 Related work

Several works [17, 39, 43] have presented remarkable unpaired image-to-image translations, using a framework commonly referred to as CycleGAN. The key idea is that the ill-posed conditional generative process can be regularized by a cycle-consistency constraint, which forces the translation to perform a bijective mapping. The cycle constraint has become a popular regularization technique for unpaired image-to-image translation. For example, the UNIT framework [24] assumes a shared latent space between the domains and enforces the cycle constraint in the shared latent space. Several works were developed to extend the one-to-one mapping to many-to-many mapping [25, 15, 20, 2]. These methods decompose the encoding space to shared latent space, representing the domain invariant content space, and domain specific style space. Therefore, many translations can be achieved from a single content code by changing the style code of the input image.

Many translation methods share the inability to translate high-level semantics, including different shape geometry. This type of translation is usually necessary in the case of transfiguration, where one aims to transform a specific type of objects without changing the background. Lee et al. [20] and Mejjati et al. [27]

learn an attention map and apply translation only on the the foreground object. However, both methods only improve translations that do not deform shapes. Gokaslan et al. [8] succeed in preforming several shape-deforming translations by several modifications to the CycleGAN framework, including using dilated convolutions in the discriminator. However, they do not demonstrate strong shape deformations, such as zebras to elephants or giraffes, as we show in Section 4.

Some works [21, 28] assume some kind of segmentation is given, and use this segmentation to guide shape deformation translation. However, such segmentation is hard to achieve. In a recent work, Wu et al. [35] disentangle the input images to geometry and appearance spaces, relying on high intra-consistency, and learn to translate each of the two domains separately. However, the variation of geometry and appearance of in-the-wild images is too large to be disentangled successfully[3].

Contrary to the above works, our work leverages a pre-trained network and the translation is applied directly on deep feature maps, thus being guided by high-level semantics. Several image-to-image methods, such as [38, 4, 16], also incorporate such pre-trained networks, though usually, only as perceptual loss, constraining the translated image to remain semantically close to the input image. Differently, Sungatullina et al. [31] incorporate pre-trained VGG features into the discriminator architecture, to assist in the discrimination phase. Wu et al. [36] use VGG-19 as a fixed encoder in the translation, where only the decoder is learned. Upchurch et al. [33] present the only method, to our knowledge, that actually translates deep features between two domains. However, the translation is not learned, but defined by simply interpolating between the deep features, which restricts the scope of method to highly aligned domains. In another context, Yin et al. [40] train an autoencoder to embed point clouds, and perform translation in the learned embedding. In contrast, we utilize semantics to preform the translation in the much more difficult scenario of images.

Our work shares some similarities with Huang et al. [14], who suggest using a generative adversarial model [9] in a coarse-to-fine manner with respect to a pre-trained encoder. The generation process begins from the deepest features and then recursively synthesizes shallower layers conditioned on the deeper layer, until generating the final image. This method was only applied on small encoders and low resolution images and was not explored for very deep and semantic encoding neural networks such as VGG-19 [30].

Deep image analogies [22] transfer visual attributes between semantically similar images, by feed-forwarding them through a pre-trained network. Their work does not train a generative model; nonetheless, they create new deep features by fusing content features from one image with style features of another. Similarly, Aberman et al. [1] synthesize hybrid images from two aligned images by selecting the dominant deep feature activations.

[3] Unfortunately, at the time of this submission the authors of [35] were unable to release their code or train their network on the datasets presented in our paper.

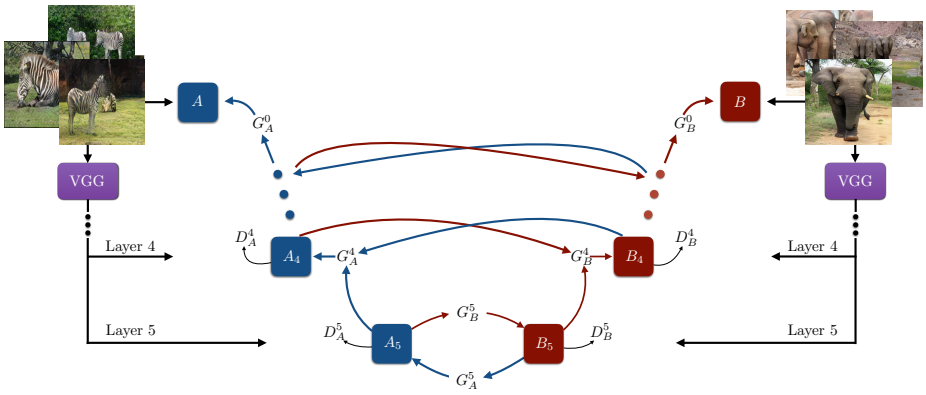

Fig. 2: Translation architecture. We translate between domains A and B starting from the deepest feature maps A_5 and B_5, which encode the highest level semantic content of the images. Translation proceeds from deeper to shallower feature maps until reaching the image itself. The feature maps are extracted by feed forwarding every image through the pre-trained VGG-19 network and sampling five of its layers. The translation of each layer is learned individually, conditioned on the translation result of the next deeper layer (except the deepest layer, whose translation is unconditional).

3 Method

Our general setting is similar to that of previous unpaired image-to-image translation methods. Given images from two domains, A and B, our goal is to learn to translate between them. However, unlike other image-to-image translation methods, we perform the translation on the deep features extracted by a pre-trained classification network, specifically VGG-19 [30].

The translation is carried out in a deep-to-shallow (coarse-to-fine) manner, using a cascade of pairs of translators, one pair per layer. The entire architecture used to train the translators is shown schematically in Fig. 2, while Fig. 3 illustrates the test-time translation (inference) process. Once the deepest feature map has been translated, we translate the next (shallower and less semantic feature map), conditioned on the translated deeper layer. In this manner, the translation of the shallower map preserves the general structure of the translated deeper one, but adds finer details, which are not encoded in the deeper feature maps. We repeat this procedure until the original image level is reached. Below we describe the training and the inference processes in more detail.

Pre-processing: We extract high-level semantic features from input images from both domains, A and B, by feed-forwarding the images through the pre-trained VGG-19 [30] network. Next, we sample five of the resulting deep feature maps, specifically `conv_i_1` (`i` $= 1, 2, 3, 4, 5$), where each map has progressively coarser spatial resolution, but a larger number channels. We denote the i-th sampled feature map for image $a \in A$ as a_i. Since propagation through the pre-trained

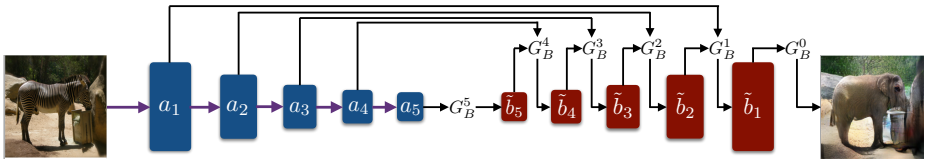

Fig. 3: Translation at test time. The input (left) is fed forward through VGG-19, yielding a set of deep feature maps. Then, we translate each feature map, starting from a_5. The final result is obtained from the shallowest translated map by feature inversion.

VGG-19 network may yield features in any range, while the range of the synthesized features is usually known, we first normalize each channel, of every layer i, by calculating its mean and standard deviation across the domain and clamp the normalized feature values to the range of $[-1, 1]$. While the clamping is a potentially harmful irreversible operation, we did not observe any adverse effect on the results. We use A_i (B_i) to denote set of all normalized deep feature maps of level i, extracted from images in domain A (B).

Inference: We perform the translation in a coarse-to-fine fashion. Thus, the translator from domain A to B, actually consists of a sequence of translators $\left\{G_B^5, G_B^4, \ldots, G_B^1\right\}$, where each translator is responsible for translating the i-th feature map layer a_i, from A_i to B_i conditioned on the previously translated deeper layer $\tilde{b}_{i+1}$ (except for the deepest layer translator G_B^5, which is unconditioned). Finally, G_B^0 uses feature inversion to convert $\tilde{b}_1$ to obtain the translated image. The translators G_A^i from domain B_i to A_i are defined symmetrically. The entire inference pipeline is shown in Fig. 3.

Feature inversion: In all the results we show, e.g., Fig. 1, we visualize the output of the various translators by pre-training a deep feature inversion network (per domain), for each layer $i = 1, \ldots, 5$, following [5]. The network aims to reconstruct the original image given the feature map of a specific layer, regularized by adversarial loss so that the reconstructed image would lie in the manifold of natural images. For more details we refer the reader to [5]. The specific settings used in our implementation are elaborated in the supplementary materials.

Deepest layer translation: We begin by translating the deepest feature maps, encoding the highest-level semantic features, i.e., A_5 and B_5, hence, our problem is reduced to translating high-dimensional tensors. Our solution builds upon the commonly used CycleGAN framework [43]. Specifically, we use the three losses proposed in [43]. First, in order to generate deep features in the appropriate domain, we utilize an adversarial domain loss $\mathcal{L}_{adv}$. We simultaneously train two translators G_A^5, G_B^5 which try to fool domain-specific discriminators, D_A^5, D_B^5 (for domains A_5, B_5, respectively). However, differently from [43] and other image translation methods [15, 28], we have found LSGAN [26] not to be well-suited for our task, leading to mode collapse or convergence failures. Instead, we found

WGAN-GP [10] more effective, thus, the adversarial loss for translation from X to Y is defined as

$$\mathcal{L}_{adv}\left(G_Y, D_Y, X, Y\right) = \mathop{\mathbb{E}}_{x\sim\mathbb{P}_X}\left[D_Y\left(G_Y\left(x\right)\right)\right] - \mathop{\mathbb{E}}_{y\sim\mathbb{P}_Y}\left[D_Y\left(y\right)\right] + \lambda_{gp}\mathop{\mathbb{E}}_{\hat{y}\sim\mathbb{P}_{\hat{Y}}}\left[\left(\left\|\nabla D_Y\left(\hat{y}\right)\right\| - 1\right)^2\right], \quad (1)$$

where $G_Y : X \to Y$ is the translator, D_Y is the target domain discriminator, $\lambda_{gp} = 10$ in all our experiments, and $\mathbb{P}_{\hat{Y}}$ is defined by uniformly sampling along straight lines between $\tilde{y} \sim G\left(\mathbb{P}_X\right)$ and $y \sim \mathbb{P}_Y$. For more details we refer the reader to [10].

Second, for regularizing the translation to a one-to-one mapping, we add the cycle consistency loss,

$$\mathcal{L}_{cyc}(G_X, G_Y, X, Y) = \mathop{\mathbb{E}}_{x\sim\mathbb{P}_X}\left\|G_X\left(G_Y\left(x\right)\right) - x\right\| + \mathop{\mathbb{E}}_{y\sim\mathbb{P}_Y}\left\|G_Y\left(G_X\left(y\right)\right) - x\right\|, \quad (2)$$

where $\|\cdot\|$ stands for the L_1 norm.

Finally, as in [43], we have found it helpful to use an identity loss, which guides the networks to preserve common high level features,

$$\mathcal{L}_{idty}(G_X, G_Y, X, Y) = \mathop{\mathbb{E}}_{x\sim\mathbb{P}_X}\left\|G_X\left(x\right) - x\right\| + \mathop{\mathbb{E}}_{y\sim\mathbb{P}_Y}\left\|G_Y\left(y\right) - y\right\|. \quad (3)$$

The entire loss combines these components as follows

$$\begin{aligned}\mathcal{L}^5 =& \mathcal{L}_{adv}\left(G_B^5, D_B^5, A_5, B_5\right) + \mathcal{L}_{adv}\left(G_A^5, D_A^5, B_5, A_5\right) \\ &+ \lambda_{cyc}\mathcal{L}_{cyc}(G_A^5, G_B^5, A_5, B_5) + \lambda_{idty}\mathcal{L}_{idty}(G_A^5, G_B^5, A_5, B_5), \end{aligned} \quad (4)$$

where λ_{cyc} and λ_{idty} were set to 100 in all our experiments.

Coarse to fine conditional translation: Consider two successive layers, $a_i \in A_i$ and $a_{i+1} \in A_{i+1}$, where the latter has already been translated, yielding $\tilde{b}_{i+1}$ as the translation outcome (see Fig. 3). We next perform the translation of the layer a_i to yield $\tilde{b}_i$, using the translator G_B^i, conditioned on $\tilde{b}^{i+1}$. Note that G_B^i is effectively a function of all the previously translated layers.

The architecture of G_B^i is schematically shown in Fig. 4. Since shallower layers encode less of the semantic content of the image, it is more difficult to learn how they should be deformed, and thus they are used to transfer "style", while the "content" comes from the already translated deeper layer. Inspired by [15], we add an adaptive instance normalization (AdaIN) [13] component, whose parameters are learned from the current layer. Thus, several layers of G_B^i are normalized according to the AdaIN component. G_A^i, which is designed symmetrically, is learned simultaneously with G_B^i, as shown in Fig.4(a).

The loss for training these shallower translators is defined similarly to that used for training the deepest translation: it consists of adversarial, cycle consistency, and identity terms. While the adversarial loss is unconditional, similarly to (1), the cyclic loss is now conditioned: $\left\|G_A^i\left(G_B^i\left(a_i, \tilde{b}_{i+1}\right), a_{i+1}\right) - a_i\right\| +$

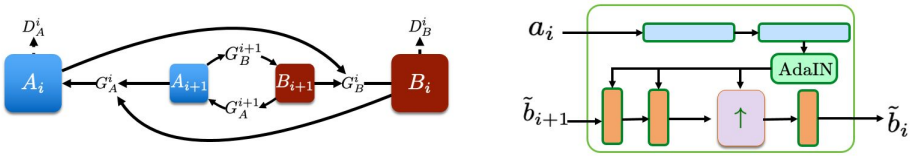

Fig. 4: Translation of layer i is conditioned on the previously translated layer $i+1$. The two translators G_A^i and G_B^i are trained simultaneously (see left figure), while the $i+1,\ldots,5$ translators are fixed. On the right we show the schematic architecture of G_B^i which has two inputs: $a_i \in A_i$ and $\tilde{b}_{i+1}$. a_i is fed-forward through several layers to yield AdaIN parameters which control the generation of $\tilde{b}_i$. Since $\tilde{b}_i$ has twice the spatial size of $\tilde{b}_{i+1}$, we add an upsampling layer marked by ↑.

$\left\|G_B^i\left(G_A^i\left(b_i,\tilde{a}_{i+1}\right),b_{i+1}\right)-b_i\right\|$, and the same conditioning is used for the identity loss: $\left\|G_A^i\left(a_i,a_{i+1}\right)-a_i\right\|+\left\|G_B^i\left(b_i,b_{i+1}\right)-b_i\right\|$.

We train the pairs of translators one layer at a time, starting from G_A^5 and G_B^5. More details regarding the implementation and the training of the translators are included in the supplementary materials.

4 Experiments

We evaluate our approach on several publicly available datasets: (1) Cat ↔ Dog faces [20], which contains 871 cat images and 1364 dog images and does not require high shape deformation; (2) Kaggle Cat ↔ Dog [6] dataset with over 12,500 images in each domain, where images may contain part of the subject or several instances; (3) MSCOCO dataset [23], specifically, zebra ↔ elephant and zebra ↔ giraffe (overall there are 1917 zebras, 2547 giraffes and 2144 elephants). These are extremely challenging datasets, and it should be noted that no previous method has used MSCOCO, without supervision in the form of segmentation.

Our deepest translators, i.e., G_A^5, G_B^5, consist of encoder-decoder structure with several strided convolutional layers followed by symmetric transpose convolutional layers. We use group normalization [37] and ReLU activation function (except the last layer, which is `tanh`). The conditional generators, consist of learned AdaIN layer, achieved by several strided convolutional layers followed by fully connected layers. The content generator has also several convolutional layers and one single transpose convolutional layer which doubles the spatial resolution (Fig. 4(right)). In practice we only train G^5, G^4, G^3, and apply feature inversion directly on the output of the latter, with negligible degradation. For the exact layer specifics we refer the reader to the supplementary materials, and to our (soon to be published) code. We train each layer for 400 epochs with a fixed learning rate of 0.0001 using the Adam optimizer [18]. On a single RTX 2080, training the entire ensemble of networks (all translators, from the deepest layer to the shallowest layer, and the final feature inversion network), takes around 2.5 days.

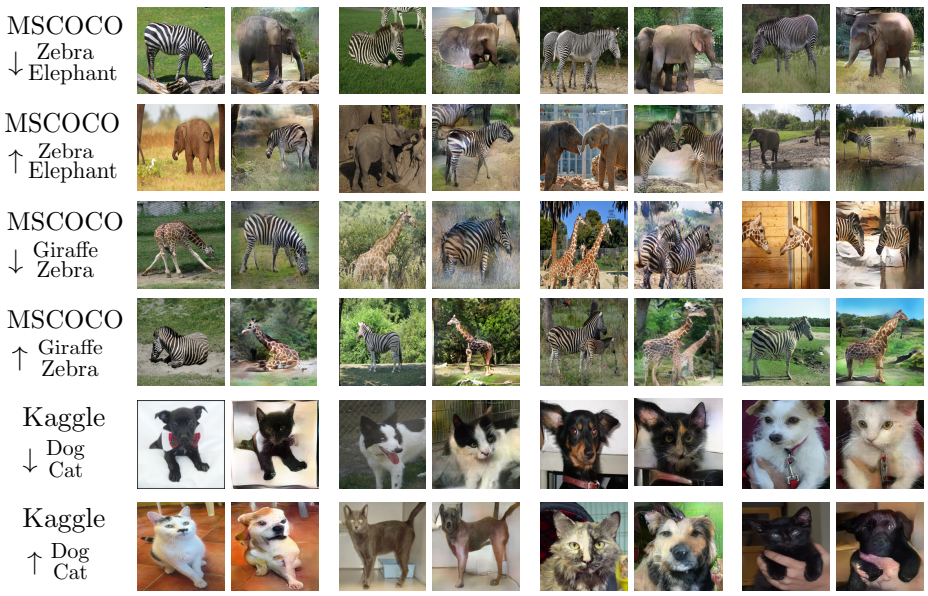

Fig. 5: Examples of challenging translation results, featuring significant shape deformations.

Several translation examples are presented in Fig. 5. Our translation is able to achieve high shape deformation. Note that our translations are semantically consistent, in the sense that they preserve the pose of the object of interest, and the number of instances is mostly preserved. Furthermore, partial occlusions of such objects, or their cropping by the image boundaries are correctly reproduced. See for example, the translations of the pairs of animals in columns 5–6. More results are provided in the supplementary materials.

4.1 Ablation study

Below, we analyze the impact of the main elements of our method.

Loss components First, we ablate each of our loss components. Fig. 6 visualizes the translation of the 5th (deepest) layer with and without cycle, identity and adversarial losses. The best result is obtained by using all of the losses, which balance each other.

Translation depth In Fig. 7 we compare between translation results using different VGG-19 layers. Evidently, shallower layers introduce more rigid spatial constraints, restricting the ability of shapes to be changed by the translation. The shallowest layer can hardly change the shape of the input image, which may explain the failure of traditional image translation methods. In Table 1, we use the common FID score [32] to show that the cascaded translation achieves better translation compared to individual layer translation. Additional results are shown in the supplementary materials.

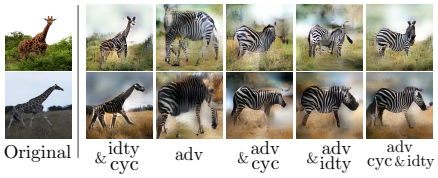

Fig. 6: Translation of the 5th (deepest) layer with different loss combinations. Using all three components yields the best result.

Fig. 7: Translation of different VGG layers, separately. Low level semantics translation fails to deform the geometry of the object.

Table 1: FID score comparison of different layer translation. Each translation was trained independently. We compare the FID scores on three datasets, measured both directions per dataset. The two directions appear side-by-side, →/←, at each cell

→/←	Layer 5	Layer 4	Layer 3	Cascaded (ours)
Cat ↔ Dog	126.93/127.53	181.90/164.42	178.13/91.71	**67.58/46.02**
Zebra ↔ Giraffe	167.62/184.37	103.41/53.36	112.43 /68.62	**67.41/39.38**
Zebra ↔ Elephant	101.26/76	105.58/57.34	166.32/113.28	**68.45/47.86**

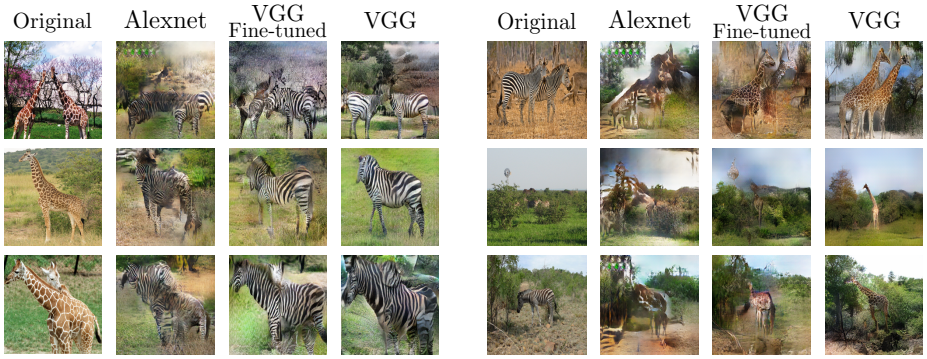

Fig. 8: Translation with different pre-trained networks. All the networks were pre-trained on ImageNet. VGG was further fine-tuned to classify between zebras and giraffes. Evidently, using this fine-tuned version does not improve the translation results. In addition, translation between AlexNet features fails to produce reasonable results.

Table 2: FID score comparison. We compare our FID scores against five approaches on three datasets, measured for both translation directions per dataset. The two directions appear side-by-side, →/←, at each cell

→/←	CycleGAN	MUNIT	DRIT	GANimorph	Ours
Cat ↔ Dog	125.75/94.27	159.57/108.51	153.94/139.17	139.17/134.14	**67.58**/**46.02**
Zebra ↔ Giraffe	**55.65**/58.93	238.06/60.78	59.75/54.06	98.25/120.05	67.41/**39.38**
Zebra ↔ Elephant	86.55/68.44	109.56/80.1	78.01/56.39	99.98/89.74	**68.45**/**47.86**

Type of pre-trained network While our method is conceptually agnostic to the type of feature extraction network, we rely on the assumption that the extracted features represent high-level semantics. Therefore, we chose the VGG-19 deep features, which are commonly used for image generation tasks [1, 5, 7, 22]. Nonetheless, we experimented with a fine-tuned version of VGG-19, as well as a different network architecture, as shown in Fig. 8. We first fine-tuned VGG-19 to classify between zebras and giraffes and trained our translation networks using the resulting features. As can be seen, the translation results are inferior to the results achieved by the standard VGG-19. This may be attributed to VGG-19 fixating on the unique differences between the zebra and giraffe images, unrelated to the translation, such as background. For more about the extracted features, we refer the reader to the supplementary materials. In addition, in Fig. 8, we examine a different network, AlexNet, also pretrained on ImageNet. We observe that the deepest image translation is not able to generate valid shapes of zebras or giraffes. AlexNet uses a stride of 4 in its first convolutional layer. Thus, the resulting features have less spatial encoding, especially at the deeper layers, which may explain the difficulty to invert and translate these features.

4.2 Comparison to other methods

We compare our result with several leading image-to-image translation methods, i.e., CycleGAN [43], MUNIT [15], DRIT [20] and GANimorph [8].

Quantitative comparison: In order to perform a quantitative comparison, we use the FID score [32], as reported in Table 2. Our method achieves the best FID score on five out of the six cross-domain translations for which this score was measured.

Qualitative comparison: In Fig. 9 we show several challenging translation examples. While traditional image translation methods struggle to preform translations with such drastic shape deformation, our method is able to do so thanks to its use of the pre-trained VGG-19 network.

The success of our method can also be explained and visualized by examining the translated deep features. We feed forward every image, original and translated, through the entire VGG network, extracting the last fully-connected layer (before the classification layer). We project this vector (of size 4096) to

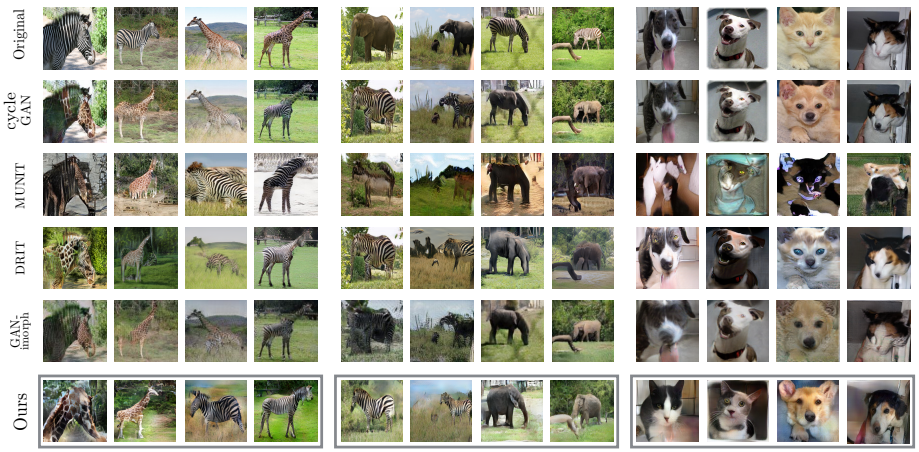

Fig. 9: Comparison to other image-to-image translation methods. The unpaired translations, from left to right, are zebra ↔ giraffe, elephant ↔ zebra, and Kaggle dog ↔ cat, where every translation has four examples, two in each direction. While previous translation methods struggle to deform the geometry of the source images, our method is able to preform drastic geometric deformation, while preserving the poses of the subjects and the overall composition of the image.

2D, using t-SNE, as shown in Fig. 10 It may be seen that the distribution of the translated vectors (in cyan) is closest to that of the target domain (in red) when using our method.

Limitations Our method achieves translations with significant shape deformation in many previously unattainable scenarios; yet, a few limitations remain. First, the background of the object is not preserved, as the background is encoded in the deep features along with the semantic parts. Also, in some cases the translated deep features may be missing small instances or parts of the object. This may be attributed to the fact that VGG-19 is generally not invertible and was trained to classify a finite set of classes. In addition, since we translate deep features, small errors in the deep translation may be amplified to large errors in the image, while for image-to-image translation method that operate on the image directly, small translation errors would typically be more local. Please note that, similarly to CycleGAN and GANimorph, our translation is deterministic.

4.3 Unconditional generation via deep feature synthesis

The expressive power of deep features can also be leveraged by unconditional generative models that synthesize the deep features, rather then generating the images directly. Specifically, we demonstrate that such generative models are able to compete with state-of-the-art synthesis networks, especially with respect to higher-lever semantics. We train a variational auto encoder (VAE) [19] to

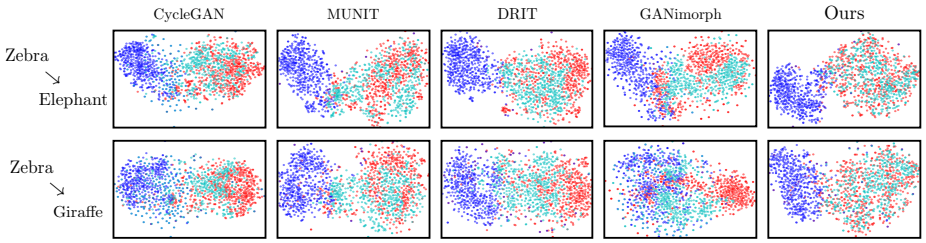

Fig. 10: Comparison of the deepest latent spaces (5th layer), projected using t-SNE. The latent space of the source domain is in blue, and the target domain is in red. The distribution of the translation results (in cyan) is most similar to that of the target domain when using our method.

generate the `conv_5_1` feature maps of zebras (using feature maps extracted by VGG-19 pretrained on ImageNet, as our training data). We then synthesize shallower layers in a cascaded fashion, using a process similar to the one described in Section 3 (for more details please refer to the supplementary materials). We refer to the resulting generative model as DEEP-VAE. As shown in Fig. 11, the images generated by DEEP-VAE are not blurry, a phenomenon ordinary VAEs are notoriously known for. We compare our DEEP-VAE to DFC-VAE [12], which uses a perceptual loss for reconstruction, and to VQ-VAE-2 [29], a state-of-the-art VAE synthesis module, which learns a multi-categorical distribution over a learned dictionary elements. As shown in Fig. 11, DFC-VAE fails to produce clear and sharp images, while many of the VQ-VAE-2 results do not contain the main semantic attribute (zebras, giraffes or elephants) at all. This is also evident in the FID scores, shown in Table 3. In order to further demonstrate the generative power of deep features, we also show several examples of latent interpolation in Fig. 12. More results are reported in the supplementary materials.

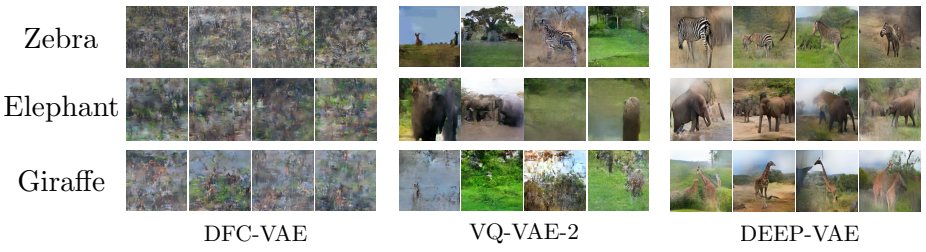

Fig. 11: Synthesis quality comparison. While DCF-VAE is trained using a perceptual loss, it is unable to produce realistic results. VQ-VAE-2 is able to generate higher quality images, however these images rarely contain the main semantic content of the training dataset, i.e. zebras, giraffes and elephants. Our method produces good quality images with the structure of the animal evident in almost all of the generated images.

Table 3: FID score comparison for VAE synthesis

Dataset	DFC-VAE	VQ-VAE-2	DEEP-VAE
Zebra	324	154.41	**57.66**
Elephant	347.93	267.32	**80.2**
Giraffe	346.47	254	**108.02**

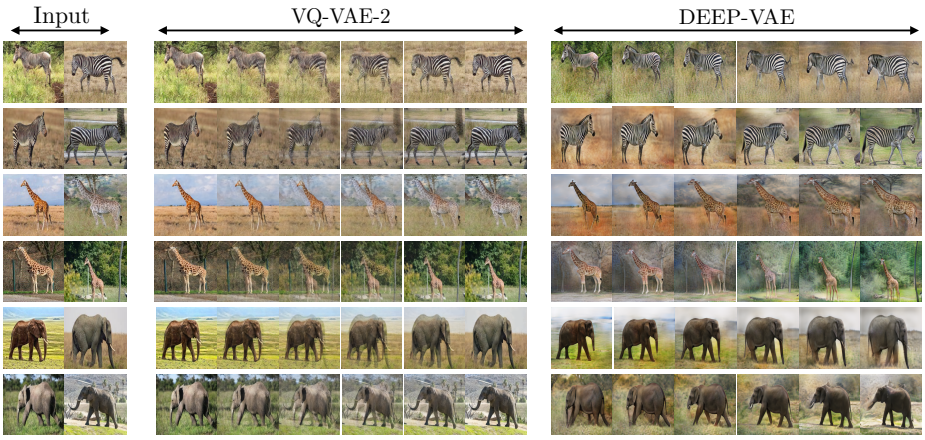

Fig. 12: Latent interpolation between deep features. Two input images are encoded by a trained VAE. Uniform interpolation is preformed between the two encodings, and the decoded result is shown for both VQ-VAE-2 and DEEP-VAE. While DEEP-VAE has a very simple architecture it competes with the state of the art VQ-VAE-2 w.r.t reconstruction and yield interpolation results without ghosting artifacts.

5 Conclusions

Translating between image domains that differ not only in their appearance, but also exhibit significant geometric deformations, is a highly challenging task. We have presented a novel unpaired image-to-image translation scheme that operates directly on pre-trained deep features, where local activation patterns provide a rich semantic encoding of large image regions. Thus, translating between such patterns is capable of generating significant, yet semantically consistent, shape deformations. In a sense, this solution may be thought of as transfer learning, since we make use of features that were trained for a classification task for an unpaired translation task. We have also demonstrated the potential of such transfer learning in the context of unconditional image generation. In the future, we would like to continue exploring the applications of powerful pre-trained deep features for other challenging tasks, possibly in different domains, such as videos, sketches or 3D shapes.

References

1. Aberman, K., Liao, J., Shi, M., Lischinski, D., Chen, B., Cohen-Or, D.: Neural best-buddies: Sparse cross-domain correspondence. ACM Transactions on Graphics (TOG) **37**(4), 69 (2018)
2. Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A.: Augmented CycleGAN: Learning many-to-many mappings from unpaired data. arXiv preprint arXiv:1802.10151 (2018)
3. Benaim, S., Wolf, L.: One-sided unsupervised domain mapping. In: Advances in neural information processing systems. pp. 752–762 (2017)
4. Di, X., Sindagi, V.A., Patel, V.M.: GP-GAN: Gender preserving GAN for synthesizing faces from landmarks. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 1079–1084. IEEE (2018)
5. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in neural information processing systems. pp. 658–666 (2016)
6. Elson, J., Douceur, J.J., Howell, J., Saul, J.: Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: ACM Conference on Computer and Communications Security (2007)
7. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
8. Gokaslan, A., Ramanujan, V., Ritchie, D., In Kim, K., Tompkin, J.: Improving shape deformation in unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 649–665 (2018)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: Advances in Neural Information Processing Systems. pp. 5767–5777 (2017)
11. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017)
12. Hou, X., Shen, L., Sun, K., Qiu, G.: Deep feature consistent variational autoencoder. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1133–1141. IEEE (2017)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
14. Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5077–5086 (2017)
15. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 172–189 (2018)
16. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: Wespe: weakly supervised photo enhancer for digital cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 691–700 (2018)

17. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1857–1865. JMLR. org (2017)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 35–51 (2018)
21. Liang, X., Zhang, H., Lin, L., Xing, E.: Generative semantic manipulation with mask-contrasting gan. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 558–573 (2018)
22. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. arXiv preprint arXiv:1705.01088 (2017)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
24. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems. pp. 700–708 (2017)
25. Ma, L., Jia, X., Georgoulis, S., Tuytelaars, T., Van Gool, L.: Exemplar guided unsupervised image-to-image translation. arXiv preprint arXiv:1805.11145 (2018)
26. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2794–2802 (2017)
27. Mejjati, Y.A., Richardt, C., Tompkin, J., Cosker, D., Kim, K.I.: Unsupervised attention-guided image-to-image translation. In: Advances in Neural Information Processing Systems. pp. 3693–3703 (2018)
28. Mo, S., Cho, M., Shin, J.: Instagan: Instance-aware image-to-image translation. arXiv preprint arXiv:1812.10889 (2018)
29. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: Advances in Neural Information Processing Systems. pp. 14837–14847 (2019)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
31. Sungatullina, D., Zakharov, E., Ulyanov, D., Lempitsky, V.: Image manipulation with perceptual discriminators. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 579–595 (2018)
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
33. Upchurch, P., Gardner, J., Pleiss, G., Pless, R., Snavely, N., Bala, K., Weinberger, K.: Deep feature interpolation for image content changes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7064–7073 (2017)
34. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
35. Wu, W., Cao, K., Li, C., Qian, C., Loy, C.C.: TransGaGa: Geometry-aware unsupervised image-to-image translation. arXiv preprint arXiv:1904.09571 (2019)

36. Wu, X., Shao, J., Gao, L., Shen, H.T.: Unpaired image-to-image translation from shared deep space. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 2127–2131. IEEE (2018)
37. Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
38. Xu, J., Sun, X., Zeng, Q., Ren, X., Zhang, X., Wang, H., Li, W.: Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. arXiv preprint arXiv:1805.05181 (2018)
39. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. In: Proc. IEEE ICCV. pp. 2849–2857 (2017)
40. Yin, K., Chen, Z., Huang, H., Cohen-Or, D., Zhang, H.: Logan: Unpaired shape transform in latent overcomplete space. arXiv preprint arXiv:1903.10170 (2019)
41. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. arXiv preprint arXiv:1806.03589 (2018)
42. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)
43. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proc. IEEE ICCV. pp. 2223–2232 (2017)