Computational Visual Media
https://doi.org/10.1007/s41095-021-0216-x Vol. 7, No. 4, December 2021, 453–466
Research Article

Image resizing by reconstruction from deep features

Dov Danon[1,*] (✉), **Moab Arar**[1,*], **Daniel Cohen-Or**[1], **and Ariel Shamir**[2]

Abstract Traditional image resizing methods usually work in pixel space and use various saliency measures. The challenge is to adjust the image shape while trying to preserve important content. In this paper we perform image resizing in feature space using the deep layers of a neural network containing rich important semantic information. We directly adjust the image feature maps, extracted from a pre-trained classification network, and reconstruct the resized image using neural-network based optimization. This novel approach leverages the hierarchical encoding of the network, and in particular, the high-level discriminative power of its deeper layers, that can recognize semantic regions and objects, thereby allowing maintenance of their aspect ratios. Our use of reconstruction from deep features results in less noticeable artifacts than use of image-space resizing operators. We evaluate our method on benchmarks, compare it to alternative approaches, and demonstrate its strengths on challenging images.

Keywords image retargeting; reconstruction; deep seam carving; image resizing

1 Introduction

The media resizing problem had been widely studied in the last decade and many content-aware methods have been developed [1–13]. The main objective of these methods is to change the size of the input while maintaining the appearance of important regions such as salient objects, and reducing visual artifacts. These two objectives can be seen as two quality measures that are sometimes contradicting. The first measures how semantically close the resulting image is to the original by preserving its important parts, and the second measures the resemblance of the result to a natural image by reducing artifacts (see Ref. [14]).

Most resizing techniques first employ a saliency detection method to decide which regions of the image are more important. Then, an image resizing operator is used to create the resized image while preserving these regions, hoping to introduce few artifacts. Both of these steps are still challenging. Firstly, common saliency measures account mostly for low-level features, while disregarding important high-level semantics. Secondly, current resizing operators do not directly account for the second quality measure of maintaining the natural look of the resulting image.

In this work we present *deep network resizing* (DNR) as a method that deals with the two aforementioned challenges using neural networks. First, we exploit the ability of pre-trained networks to analyze and encode both low-level and high-level features to identify important parts of the image. In addition, we employ a back-propagation aided optimization method to directly preserve both the structures of important regions and the natural appearance of the result. This results in a reduction in artifacts compared to those arising in traditional approaches, and integrates analysis and synthesis based on neural networks in an image resizing technique.

The key idea of DNR is that instead of applying image resizing operators to the pixels of the image, they are applied in feature space, to the feature maps of deep layers of a pre-trained convolutional neural network (see Fig. 1). This draws content removal to regions of the image that are semantically less important. We show that DNR discards insignificant parts, which in turn, preserves the semantic encoding

* Dov Danon and Moab Arar contributed equally to this work.
1 Tel Aviv University, Tel Aviv, 69978, Israel. E-mail: D. Danon, dovdanon@post.tau.ac.il (✉); M. Arar, moabarar@mail.tau.ac.il; D. Cohen-Or, dcor@tau.ac.il.
2 The Interdisciplinary Center Herzliya, Herzliya, 4610101, Israel. E-mail: arik@idc.ac.il.
Manuscript received: 2021-01-28; accepted: 2021-02-25

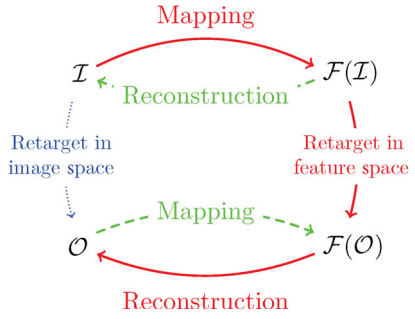


Fig. 1 Conventional resizing approaches act in image space (blue arrow), while our deep-resizing approach (red) applies resizing in the semantic feature space. We map image $\mathcal{I}$ to feature maps $\mathcal{F}(\mathcal{I})$ in feature space using a CNN. Then, we resize in feature space to create $\mathcal{F}(\mathcal{O})$. Lastly, we use back propagation optimization to reconstruct $\mathcal{O}$. In other words, instead of reconstructing the original image $\mathcal{I}$ from $\mathcal{F}(\mathcal{I})$ (green arrow), we reconstruct the resized image $\mathcal{O}$ from $\mathcal{F}(\mathcal{O})$, which is the hypothetical mapping of $\mathcal{O}$ to feature space.

of the input image. The operators we demonstrate our approach with are seam carving [1] in combination with warping.

Finally, after the image is reconstructed by optimization we perform a refinement step. In this step, a grid-sampler layer is used, allowing only changes in the mapping of pixels and not their color, while optimizing using the same objective. This step increases the natural appearance of the resulting image, by further reducing artifacts.

Our main contributions are:

- utilizing the semantic guidance of deep layers of a CNN for image importance in resizing,
- applying seam-carving in feature-space instead of image-space,
- reducing artifacts in reconstructed images by optimization using grid-sampling, and
- deep network resizing, a method for image resizing using neural networks.

2 Related work

2.1 Image processing techniques

Considerable work on content-aware media retargeting has been carried out in the field of image processing, and it is common to classify it into *discrete methods* [1, 3, 5, 8] and *continuous methods* [2, 4, 6, 7, 9, 10] (refer to Refs. [15, 16] for comprehensive coverage of content-aware retargeting).

2.1.1 Discrete methods

Seam carving was introduced by Avidan and Shamir [1], performing retargeting by repeatedly inserting or removing connected paths of pixels called *seams*, passing through low importance regions of the image. Later, Rubinstein et al. [3] improved seam carving using a look-forward energy map, which measures the amount of energy introduced by seam removal or insertion. Pritch et al. [5] introduced shift-maps for pixel re-arrangement, and formulated a graph-labeling problem for various image editing applications, including retargeting. Rubinstein et al. [8] combined different retargeting operators by finding sequences in a multidimensional space of retargeting operations on the input media.

2.1.2 Continuous methods

Wolf et al. [2] introduced a map that is determined by three importance measures in order to devise a system of linear equations that defines a mapping of source pixels into their corresponding location in the target image. Wang et al. [4] computed a deformed mesh-grid by assigning a scale factor for each quad in the grid. They proposed two penalties that encourage their solution to linearly scale quads of high-importance and allow higher deformation of low-importance quads. Krähenbühl et al. [7] used an energy map, consisting of many automatic constraints and user defined constraints on key frames, in order to compute non-uniform pixel accurate warping on video streams. Guo et al. [6] defined a saliency-based triangle mesh representation, and used a constrained mesh parametrization problem to compute the retargeting solution. Wu et al. [9] detected symmetric parts in the image and then applied summarization operations to the symmetric regions and warping to non-symmetric parts. Panozzo et al. [10] used an axis-aligned representation which reduces optimization complexity when using a 2D parametric representation of mesh deformation. The authors [10] later found the deformation parameters by solving a simple quadratic problem with linear constraints.

2.2 Deep learning techniques

The superiority of neural networks in solving computer vision tasks, including image recognition [17–20], segmentation, and detection [21–24], has already been established in the last few years.

One possible approach using deep learning for retargeting would be to gather as a training set pairs of original and retargeted images and use supervised learning. However, it is very difficult to gather such a

(a) (b) (c) (d) (e) (f)

Fig. 2 Given an input image (a), our deep network resizing method first adjusts the size of the feature maps of a deep neural network (b), while protecting important semantic regions, and then reconstructs a retargeted image using iterative optimization (c–f). Note how starting from a linear scaled image (c), the iterations manage to reconstruct the shape of the bicycle (d–f), which is the main semantic object in the image, while minimizing artifacts.

set as each image must support numerous retargeting sizes and in fact, there are no ground-truth results or methods to use. Even manual retargeting by different artists often produces different results.

Deep features were previously incorporated in image retargeting. Liu et al. [25] extracted salient patches to mimic the human gaze shifting path when viewing a scene. These patches are then used to construct a deep feature representation of the input image, which are used in a grid-based retargeting approach. Song et al. [26] adjusted photos to square format by using a deep multi-operator, which consists of scaling, cropping, and seam carving. However, human intervention is required. Kajiura et al. [27] used reinforcement learning to find the optimal retargeting order when using a multi-operator approach. In other work, Esmaeili et al. [28] proposed an automatic thumbnail generation network that does not utilize a saliency map. Unlike our method, their method generates the final image by cropping the original image, which can be limiting in extreme retargeting cases.

Cho et al. [11] proposed a weakly and self supervised learning method for image retargeting. It uses the semantic encoding of pre-trained networks and a decoder that produces an attention map. The attention map is then combined with a shift-layer in order to obtain the retargeted image. Unlike our DNR approach, Cho et al. trained their network on a given dataset, where the objective is to minimize structural damage while maintaining the detection score of the image, as given by the pre-trained CNN. In contrast, DNR performs analysis per input-image, and presents a solution that uses the strengths of deep learning both in understanding image semantics, and in correcting the resulting images. DNR utilizes different retargeting operators to produce a feature representation of the target image (see comparison in Fig. 12).

Independently from our work, Lin et al. [29] recently proposed to perform retargeting in feature space. However, there are two fundamental differences between their work and ours. Firstly, they preform retargeting by sampling columns of deep feature maps at a constant rate, while we can combine several deep retargeting operators. Further, the authors adapt methods from Ref. [30] and perform warping on the input image using PatchMatch [31], while our image is reconstructed via a pure synthesis procedure (see comparison in Fig. 13), and we further apply grid-sampling to reduce artifacts.

In a more recent paper, Shocher et al. [12] and Shaham et al. [13] proposed a generative adverserial network (GAN) method for synthesizing images that can be considered a type of retargeting. The authors learn the patch distribution of the input image, and use this to generate images with similar patch statistics as the input image. However, the resulting image can have a very different structure to the original image and still contain artifacts.

3 Method

3.1 Preliminaries

Conventional image resizing applies pixel manipulation to the image. In this work, we propose a new approach, where resizing is applied in feature space, and the results are mapped back into image space by reconstruction (see Fig. 1). Our key idea is to leverage deep features of a pre-trained CNN, which encode

valuable latent semantics. By applying the resizing operators in feature space, we create *target feature maps*, where semantic information is kept unharmed. To reconstruct the output image we use optimization that iteratively minimizes the difference between the target feature maps and the actual feature maps of the optimized image.

Let $\mathcal{I}$ be an input image of size (h, w). Assuming we use a pre-trained deep-network with L layers, we define the activation values of all neurons in level i applied to input $\mathcal{I}$ as the ith *feature map* $\mathcal{F}_i(\mathcal{I})$. $\mathcal{F}(\mathcal{I})$ is the set of all feature maps for $1 \leqslant i \leqslant L$:

$$\mathcal{F}(\mathcal{I}) = \{\mathcal{F}_1(\mathcal{I}), \ldots, \mathcal{F}_L(\mathcal{I})\}$$

Each feature map $\mathcal{F}_i(\mathcal{I})$ has a certain number of channels, and a spatial dimension that depends on the size of the input $\mathcal{I}$. We denote by $(h_i^{\mathcal{I}}, w_i^{\mathcal{I}}, c_i)$ the height, width, and number of channels of the ith feature map.

Given the target size (h', w'), the task of resizing in image-space is to obtain an image $\mathcal{O}$ of size (h', w'), while maintaining important regions in $\mathcal{I}$ and reducing artifacts as much as possible. The resizing task in feature-space is defined as obtaining a set of *target feature maps*:

$$\mathcal{F}' = \{\mathcal{F}'_1, \ldots, \mathcal{F}'_L\}$$

such that for each level i, $\mathcal{F}'_i$ matches the dimension of the i-th feature map $\mathcal{F}_i(\mathcal{O})$ of the resized image $\mathcal{O}$, while preserving the important regions of the original image's feature maps $\mathcal{F}_i(\mathcal{I})$. That is, the dimensions of $\mathcal{F}'_i$ are $(h_i^{\mathcal{O}}, w_i^{\mathcal{O}}, c_i)$ but it contains the most important information in $\mathcal{F}_i(\mathcal{I})$.

To obtain the actual resized image $\mathcal{O}$ we assume that $\mathcal{F}' = \mathcal{F}(\mathcal{O})$, the hypothetical mapping of $\mathcal{O}$ to feature space, and reconstruct $\mathcal{O}$ by minimizing the difference between the output feature maps and the target feature-maps using back-propagation. Since important regions in various levels are maintained in the target feature maps $\mathcal{F}'$, the reconstructed image $\mathcal{O}$ preserves them as well. Lastly, to maintain the natural appearance of the target image and reduce artifacts, we apply a grid sampler [32] that further optimizes the constructed image.

An overview of DNR is provided in Fig. 3. The input image (top left) is fed into a pre-trained CNN (top right) and its feature maps are extracted. Applying *deep resizing* operators to selected layers yields the *target feature-maps*, indicated in yellow in the figure. The target image (bottom left) is

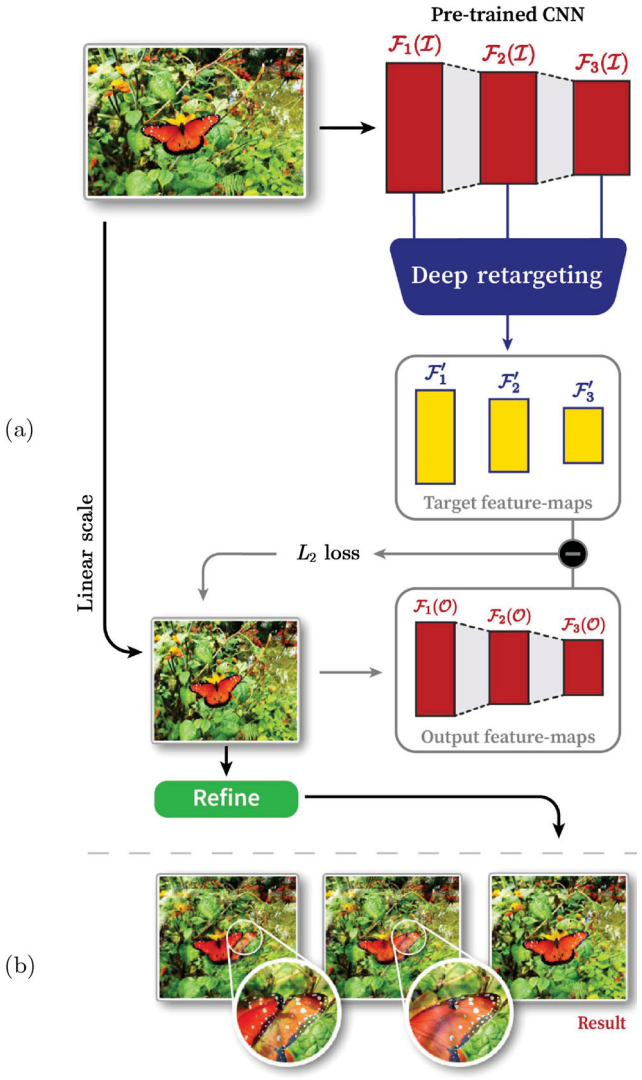


Fig. 3 Deep Network Retargeting overview. (a) The input image (left top) is fed into a pre-trained CNN (right top) and its features are extracted. Later, deep retargeting techniques are applied on the extracted features to produce the *target feature-maps* (yellow). The output image synthesis (left bottom) is achieved by iteratively minimizing the difference between the target feature maps and the actual feature maps of the output image. (b) Snapshots of the output image throughout the iterations: note how the semantic object is reconstructed.

constructed by optimization carried out using back-propagation: the result image is iteratively fed into the CNN, and an L_2-loss is computed by comparing the feature-maps of the optimized image and the target feature maps. This loss is back-propagated through the network to alter the target image during several iterations (depicted in a series of snapshots at the bottom of Fig. 3). A grid sampler further optimizes the constructed image.

In the following, we only discuss narrowing the width of the image by applying feature resizing.

Similar arguments can be extended to any other resizing target.

3.2 Feature map resizing

In our resizing method we adapt seam carving [1] and apply it to the feature maps $\mathcal{F}(\mathcal{I})$. Guided by the feature maps $\mathcal{F}(\mathcal{I})$, we conservatively utilize seam-carving while avoiding semantic regions. Doing so may lead to partially retargeted image, so we also perform a final resizing step on the reconstructed image using grid-warping [2]. This combination allows us to harness the capabilities of the two operators: seam-carving enables the removal of homogeneous unimportant regions, and grid-warping deforms regions according to their importance.

3.2.1 Deep seam carving

Seam-carving in image-space finds vertical seams as minimal one-pixel wide connected-paths using some importance map of the input image. Removing one vertical seam results in reducing the image's width by one pixel. Therefore, multiple vertical seams are removed to reach the desired width for the output image.

We extend the seam-carving algorithm by defining seam-carving on a feature-map instead of an image. Firstly, instead of removing pixels from an image we remove neurons from the CNN layer of the feature-map. Secondly, because a feature map contains multiple channels, we define seam removal as removing all neurons of the chosen seam in the same spatial location for all channels of the feature map. Thirdly, to find minimal seams in feature-maps we use a hierarchical method to define the importance-map of each layer. Starting from the deepest, lowest resolution, level which contains high-level semantic information, we move to shallower, higher resolution, layers that contain low-level features and refine the seams from previous layers consistently.

The basic importance-map of layer l at position (i, j) is defined as the L_2-norm of the activation of the neurons along the channel axis:

$$S_l(i,j) = \|\mathcal{F}_l(\mathcal{I})(i,j,*)\|_2 \tag{1}$$

where $*$ denotes all values along the channel axis (see Fig. 4).

We start by applying seam-carving to the deepest layer L in the hierarchy using the importance-map defined in Eq. (1). This map is useful since deep-layer neurons have higher activation in semantic regions. As we move up the hierarchy from level l to level $l-1$, we keep track of all seams that were removed from $\mathcal{F}_l(\mathcal{I})$. Let $SC_l = \{s_1, \ldots, s_n\}$ be the set of all chosen seams at level l (an example of one chosen seam is indicated in yellow in Fig. 5(a)).

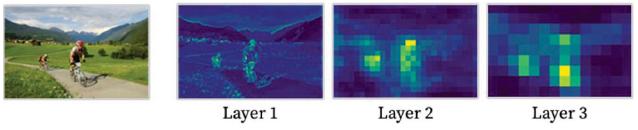
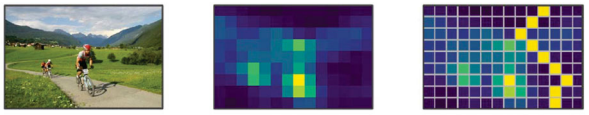


Fig. 4 The basic importance maps of Eq. (1) in different layers of the network. Yellow regions indicate high importance, while blue ones indicate low importance.

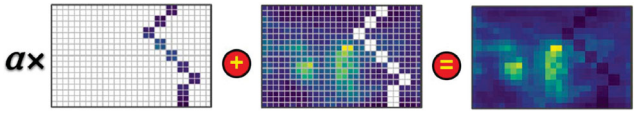
To find the minimal seams on $\mathcal{F}_{l-1}(\mathcal{I})$, we consider a modified importance map MS_{l-1} at level $l-1$ that reduces the importance of regions that are part of the receptive field of the deep seams in level l. This attracts the seams at level $l-1$ to pass through the same regions and be consistent with the seams of level l (see Fig. 5). The new map is given by the following equation:

$$MS_{l-1}(i,j) = \begin{cases} \alpha \cdot S_{l-1}(i,j), & \exists\, s \in SC_l \text{ s.t. } \mathcal{F}_{l-1}(i,j) \text{ is in the receptive field of } s \\ S_{l-1}(i,j), & \text{otherwise} \end{cases} \tag{2}$$

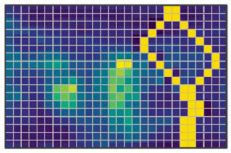
where $\alpha \in [0, 1)$ is the scaling factor. Thus, the importance map in the finer layers inherits information from deeper layers, implicitly constraining the selection of seams in the finer levels (see Fig. 6).



(a) Deep importance map and minimal seam (layer l)

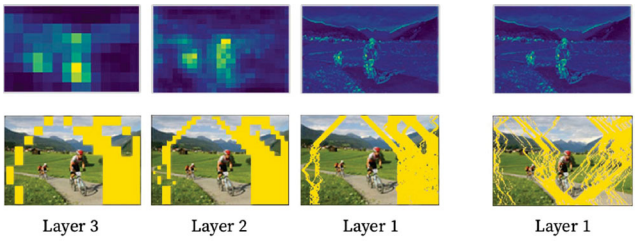


(b) Reducing the importance map as defined in Eq. (2) (for layer $l-1$) in regions where the seams pass



(c) Chosen seams at level $l-1$ using the modified importance

Fig. 5 Deep seam-carving applied hierarchically from layer l to $l-1$.



(a) DSC (b) SC

Fig. 6 Deep seam-carving vs. regular seam-carving. (a) Hierarchical deep seam-carving (DSC) applied on all three layers feature maps preserves the image's semantic information. (b) Results of original seam-carving (SC) using just the first layer feature-map as the importance map. Note how in this case, seams no longer avoid important regions.

3.2.2 Grid warping

Grid warping in image space is applied by first dividing the image into a grid of cells and then scaling each cell linearly using a different scaling factor. The scaling factors must adhere to the following two requirements. Firstly, the total width of the scaled cells must match the target width w'. Secondly, the scaling factor of each cell should be proportional to the cell's importance. The first requirement guarantees that the resulting image size will match the target size, while the second requirement ensures lower distortion in parts of higher importance.

For image width change, the initial width of each cell is given by w_G and cells are assigned a *scaling factor*, $\sigma_{i,j} \in [0,1]$, which specifies by how much each cell's width is to be decreased. The actual resizing is applied using linear scaling: the width of cell (i,j) is reduced by multiplying it by the scaling factor $\sigma_{i,j}$. In practice, it is useful to perform grid warping for width change by splitting the image into column-cells, defining only one cell in each column and one scaling factor σ_i. Otherwise, different cells in the same column may be distorted differently, which may lead to jittery results.

To define the importance value μ_i of each column-cell i, we aggregate the importance-maps calculated by Eq. (1) of all layers from the deepest to the first layer by up-sampling the deeper layer maps to fit the size of the image. The values are then normalized to define the scaling factors as

$$\sigma_i = \frac{w'}{w_G} \frac{\mu_i}{\sum_i \mu_i} \tag{3}$$

3.2.3 Deep multi-operator

The combination of deep seam-carving and grid-warping is done by preventing deep seam-carving from removing seams with semantic content. To achieve this, we terminate seam removal once the next seam's total importance is above a given threshold. However, we keep the same ratio of removed seams to the original width of the feature map in all layers, meaning that different numbers of seams are removed in each layer. Once deep seam-carving terminates, and the image is reconstructed, we apply grid-warping to the intermediate resulting image to produce the final output at the desired size.

3.3 Image reconstruction

Previous works show how to use a pre-trained CNN to synthesize images using back-propagation, for example to create images in different styles [33]. We adopt this approach, and use optimization to map back the target feature-maps into image-space to obtain the resized output image. We use the target feature-maps to reconstruct our desired output by iteratively applying back-propagation to change the values of the image by optimization. Note that what we call the output image is in fact the input image to the network.

Our initial output image $\mathcal{O}$ is set to be a uniform 1D linearly scaled version of the input image $\mathcal{I}$ (see the Electronic Supplementary Material (ESM) for comparison to other initialization methods, including random noise and a seam carved image). This allows the optimization to fix the distortion created by linear scaling, and to re-construct the desired output by iteratively reducing distortion especially in important regions of the image (see Fig. 2). Thus, we seek to update $\mathcal{O}$ by minimizing the total loss that is introduced by simple linear scaling:

$$\mathcal{L} = \sum_{i=1}^{L} \lambda_i \cdot \|\mathcal{F}_i(\mathcal{O}) - \mathcal{F}'_i\|_2 \tag{4}$$

where $\mathcal{F}'_i$ are the ith layer target feature maps, and $\mathcal{F}_i(\mathcal{O})$ are the ith layer feature maps when the output image is fed into the pre-trained CNN. Here, $\lambda_1, \ldots, \lambda_L$ are non-negative hyper-parameters to weight the contribution of each term to the total loss.

As suggested in Ref. [33], minimizing the loss in Eq. (4) using gradient descent can produce visually pleasing images. In DNR, we use the Adam optimizer [34] to solve Eq. (4).

3.4 Image refinement

Reconstruction optimization using back-propagation

changes the pixel values of the output image $\mathcal{O}$ to minimize the loss function of Eq. (4). This means that regions defined by the target feature-maps will most likely be preserved and reconstructed properly. However, some artifacts such as checkerboard patterns and noisy pixels still appear in the resulting reconstructed image. These artifacts appear because content removed from the original image causes discontinuities between better-preserved important regions, and such locations accumulate gradients more than others (similarly to artifacts created by deconvolution [35]).

We have developed a novel method that utilizes a grid-sampler layer $\mathcal{G}$ from Ref. [32] to overcome these artifacts. The grid-sampling layer learns a mapping from positions of neurons in its input to positions in the output. Here, we place such a layer as the first layer of the network, modifying the input to the network to be $\mathcal{G}(\mathcal{O})$ instead of $\mathcal{O}$ (see Fig. 7).

We use $\mathcal{G}$ only after the initial reconstruction of $\mathcal{O}$ is finalized (Section 3.3). We add the grid-sampler layer and continue to optimize by using the same loss function of Eq. (4). However, instead of changing the pixel values in $\mathcal{O}$, we keep them fixed and optimize the values of $\mathcal{G}$, the grid-sampler layer itself. In essence, this allows local shifts and interpolation of the pixels in $\mathcal{O}$, causing the optimization to push and interpolate artifacts to near-by edges and overcome unpleasant checkerboard artifacts (see Fig. 8). The final output of DNR, i.e., the resized image, is the application of the grid-sampler on the reconstructed image, i.e., $\mathcal{G}(\mathcal{O})$.

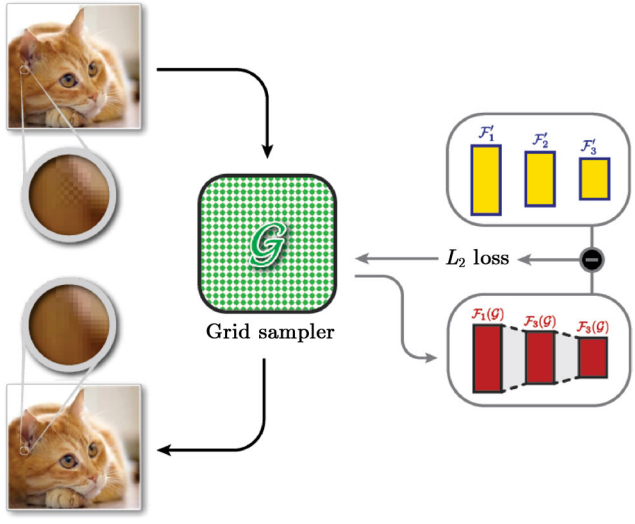


Fig. 7 The refinement procedure. The constructed image (left-top) after initial reconstruction (Section 3.3) is fed into a grid sampler optimizing the same L_2 loss function. The final output of DNR is the sampled reconstructed image (left-bottom).

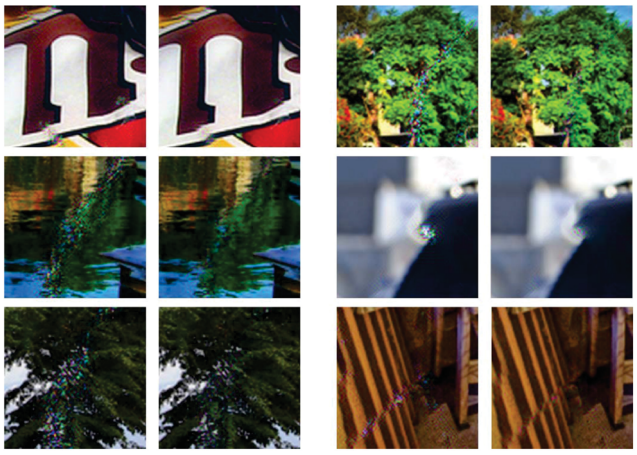


Fig. 8 Refinement examples of six patches before (left) and after (right) the refinement procedure. Note the checkerboard and other artifacts before and after the refinement (zoom-in as needed). Further results can be found in the ESM.

4 Results

4.1 Setting

In our experiments, we use VGG19 [18], which was trained on the ImageNet [36] dataset. Throughout this section, we use selected ReLU activation and Max-Pooling activation in VGG19's layers as our feature maps $\mathcal{F}_i(\mathcal{I})$. $block_i\_conv_j$ denotes ReLU activation of the jth convolution layer in block i, and $block_i\_pool$ denotes pooling activation of block i. The default configuration of our experimental results, unless otherwise stated, uses $block_1\_conv_2$, $block_2\_conv_2$, $block_3\_conv_4$, $block_4\_conv_5$, and $block_5\_pool$ as feature maps.

We always remove at least one seam in the deepest feature map, and remove more seams only if their importance is within the 20th percentile of the importance map. The value of the parameters used in the reconstruction loss (Eq. (4)) are $\lambda_1 = 1$ and for $i > 1$, $\lambda_i = 0$. The scaling factor in Eq. (2) is set to $\alpha = 0.5$. Finally, the grid size used for warping is 16.

4.2 Importance map effectiveness

The importance map used in the original seam carving algorithm [1] is based on gradient magnitude of the image. This map is often used as the base importance for many other retargeting algorithms as well. In Fig. 9, we compare the gradient-based importance

(a) (b) (c)

Fig. 9 Visualization of the importance map as seen by Deep Seam Carving. We show the gradient-based importance map (b) that is used in Seam Carving [1], and the deep-network importance map used by Deep Seam Carving (c). The deep-network importance map is more focused on semantic areas, which suggests less distortion to these regions.

map, and the deep-network importance map we use for deep seam carving. The deep-network importance map is derived by summing the importance maps used by deep seam carving. To visualize the importance map, we up-sample low-resolution maps to match the image size. As can be seen, the gradient-based importance map tends to concentrate on edges and lacks the ability to capture semantics, while our map clearly gives higher importance to semantic objects in the image.

4.3 Feature space versus image space

A possible alternative approach that also uses deep feature maps would be to apply seam carving in image space while using the feature maps as importance maps. Therefore, instead of removing seams from the feature maps, one can consider removing the same seams from the input image in order to produce the output image.

Figure 10 compares this approach to DNR based on reconstruction. As can be seen, image space retargeting leads to artifacts due to removing many seams from the same region. In contrast, in our DNR method, reconstructing the image leads to more continuous results, firstly because neighboring activations in VGG19 have overlapping receptive fields, thus affecting several output pixels in the reconstruction, and secondly, because using a CNN that trained on natural images tends to generate photo-realistic images, resulting in reduced artifacts.

(a) Image space (b) Feature space

Fig. 10 Removing seams from the input image results in discontinuous regions (a), while reconstruction using DNR produces better results (b).

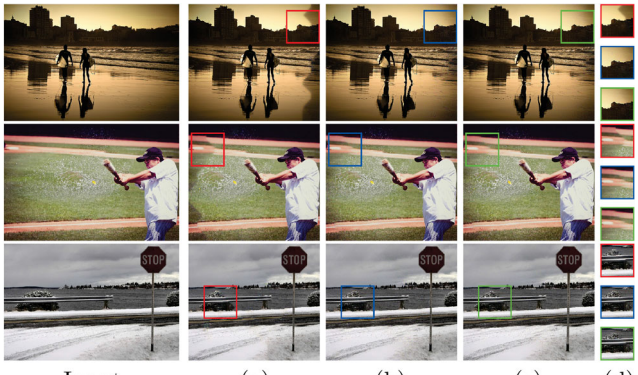
4.4 Reconstruction via deep feature-maps

In Fig. 11, we show the contributions of combining feature-maps from different levels in Eq. (4). As can be seen, using feature-maps from multiple levels improves the quality of the final image. Further details are provided in the ESM.

4.5 Visual comparison with previous methods

We compare DNR with recent deep learning based techniques [11, 29], and show some results in Figs. 12 and 13. Unfortunately, there is only limited access to the code of these methods, and we did our best to still show some side-by-side comparisons with existing results.

In addition, we use the RetargetMe benchmark [38] containing a variety of images and the results of previous retargeting operators on these images.



Input (a) (b) (c) (d)

Fig. 11 Reconstruction using multiple feature-maps from various levels. In (a) we only use shallow feature-maps in the reconstruction phase, while in (b) we use both shallow and mid-level feature maps. In (c) we show that using feature-maps from all levels produces the best outcome. For convenience, we show three patches (d) that were cropped-out from the images in (a)–(c), respectively (zoom-in if needed).

Fig. 12 50% width scale. The input images (left) are from The Pascal VOC2017 dataset [37]. We compare results of WSSDCNN [11] (middle) and DNR (right). The results are obtained by setting $\alpha = 0.2$ and employing Deep Seam Carving to perform 50% of the retargeting task. As can be seen, DNR better preserves the images subjects (see guidelines).

Fig. 13 50% width scale. (Left) Input image from RetargetMe [38], (middle) results in Ref. [29], and (right) DNR. We use $\alpha = 0.2$ and only remove seams if their importance is within the 35-percentile of the importance map. As can be seen, DNR better preserves important regions. To see this, please notice the original width of the salient subjects and compare them with each of the retargeted results (see guidelines).

We show sample results of DNR compared to Linear Scaling, Seam Carving [3], Warping [2], and Multiop [8] in Figs. 14 and 15. As can be seen, DNR better retains the aspect ratios of semantic regions compared to the other methods.

Finally, we also demonstrate our method's ability in extreme size retargeting in results in Fig. 16.

4.6 User study

To evaluate our DNR method against other alternative methods we turned to the RetargetMe benchmark [38] used to compare various methods. We conducted two forced choice tests comparing our results side-by-side to an alternative. We showed the original image before retargeting and asked the user to choose the image that best preserves the content of the original image. The order of presentation was randomly shuffled and the survey forms were randomly distributed among 112 participants.

Firstly, we chose to compare against the best performing method, SV [7]. DNR received 55.5% of the votes when compared to SV (out of 889 votes in total). Secondly, we compared against the best result obtained per image over all retargeting methods. Even in this case, our results received 52.8% of the votes (out of 956 votes in total). Counting the number of images for which users preferred our results, we found that DNR was favored for 42 images (against 25 for SV), and in 37 images (against 29 for Best).

4.7 Semantic preservation

To compare preservation of semantic details as a result of the retargeting operator, we defined a *semantic score* given by

$$\mathcal{SS}_i = \frac{\|\mathcal{F}_i(\mathcal{O})\|_2}{\|\mathcal{F}_i(\mathcal{I})\|_2} \tag{5}$$

This score compares the magnitudes of certain deep VGG-19 layer activation, before and after retargeting.

(a) (b) (c) (d) (e) (f) (g)

Fig. 14 Scaling the width of the input image by 75%. (a) Input image, (b) Linear Scale, (c) Seam Carving [3], (d) Warping [2], (e) Multiop [8], (f) SV [7], and (g) DNR (ours).

Fig. 15 Scaling the width of the input image by 50%. (a) Input image, (b) Linear Scale, (c) Seam Carving [3], (d) Warping [2], (e) Multiop [8], (f) SV [7], and (g) DNR (ours).

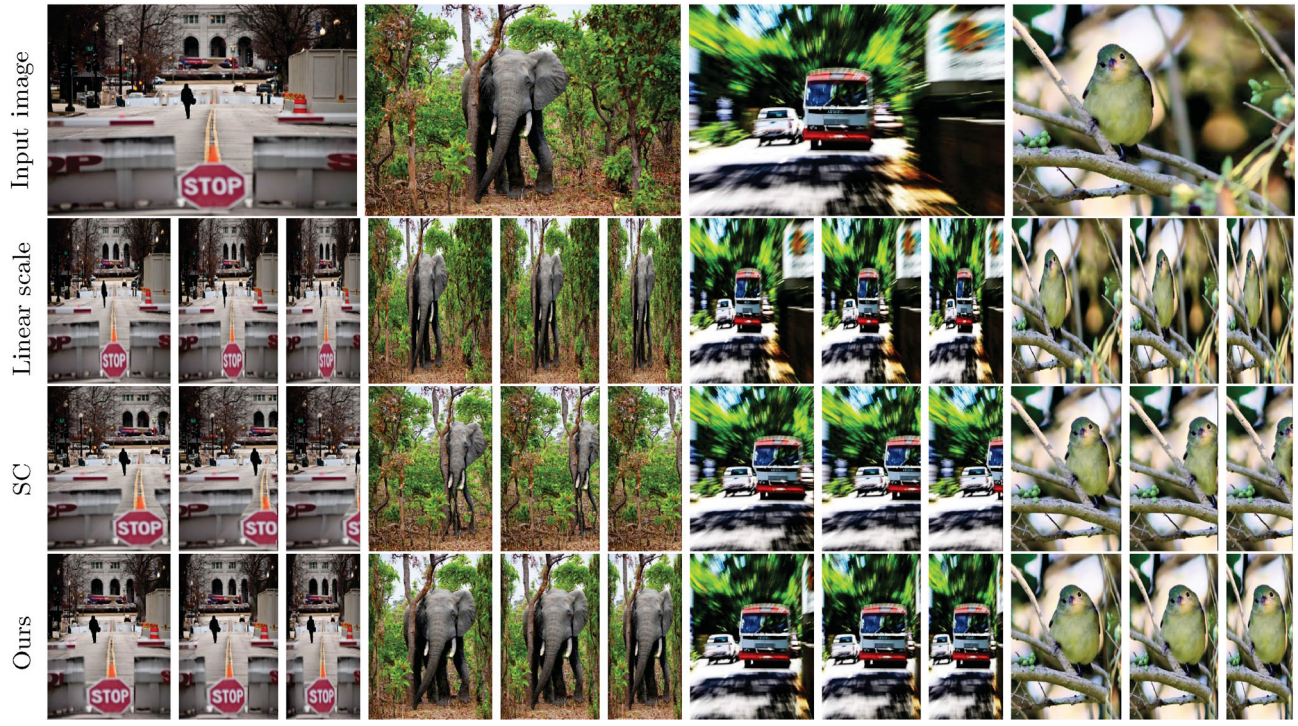


Fig. 16 Stress test of extreme retargeting on images from COCO dataset [39]. The width of the input images (first row) is scaled by 50%, 40%, and 30%. We compare Linear Scale (second row), Seam Carving [3] (third row), and our results (last row). Notice how the important subject in each image preserves its shape as much as possible.

In particular, we expect that if the retargeting operator damages semantic regions, then the score will be lower, since in this case high activation on the original image will increase the denominator $\|\mathcal{F}_i(\mathcal{I})\|_2$, while low activation on the retargeted image will diminish the numerator $\|\mathcal{F}_L(\mathcal{O})\|_2$. We used $block_5\_conv_1$ as the feature map $\mathcal{F}_i(\cdot)$ in Eq. (5). Table 1 gives the average semantic score per image for the RetargetMe benchmark and different retargeting operators. Our DNR method received the highest score.

4.8 Limitations

As with any retargeting method, artifacts may appear in the resulting image for various reasons, sometimes simply because there are no unimportant regions in

Table 1 Average Semantic Score on RetargetMe. The comparison is made between Manual Crop (CR), Linear Scale (SCL), Streaming Video (SV [7]), Seam Carving (SC [3]), and Warpping [2]. Further, we include the average semantic score computed on the best retargeted images that were chosen in RetargetMe user study (Best)

Method	CR	SCL	SV	SC	WARP	Best	**DNR**
Avg. SS	68%	68%	64%	68%	65%	65%	**70%**

an image. Still, we discuss two causes specific to our method.

Firstly, VGG19 [18] was trained for the purpose of object detection, and DNR relies on its ability to detect semantic regions and objects. However, this network does not always succeed in providing semantic information on important regions. In addition, the network detects specific features in an object and can still have low activation on different regions of important objects. All these could lead to object distortion in the final results (see Fig. 17).

Furthermore, in some cases, deep-seams constrain shallow-seams to pass through regions that are semantically unimportant. These regions may still contain edges, and removing shallow seams may cause distortion. However, since our method focuses on preserving important semantic regions, the possible artifacts tend to appear in less important regions, where these distortions are less noticeable.

Another challenge in our method is choosing a good threshold to switch from seam-carving to warping in our multi-operator scheme. In particular, we have seen cases in which further seams could have been removed while in other cases, our method removes too many seams (see Fig. 18).

Lastly, the time to produce results using DNR is still large. On average, it takes between 60–100 seconds to retarget an image of size 640×480. The speed heavily depends on the reconstruction optimization process, including the number of optimization steps and the feature maps included in Eq. (4). Our method can use similar improvement techniques for the optimization process as used in style-transfer methods (e.g., Refs. [40, 41]), which may lead to substantial speedup.

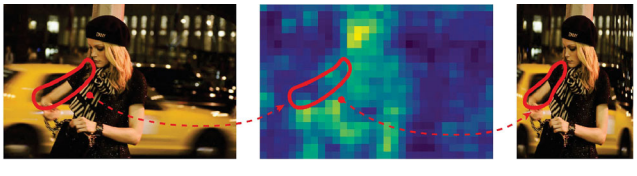


Fig. 17 When the neural network does not recognize important parts on an image (left), the corresponding deep layer has low activation (middle). In this case the hand of the woman is not detected and this leads to unwanted distortion in the output image (right).

Fig. 18 Changing the threshold determining when to switch from seam carving to warping can lead to different results. In this example, our automatic results (middle) do not preserve the aspect ratio of the sheep compared to applying only Deep Seam Carving (right).

5 Conclusions

We have presented an image retargeting technique that operates in deep layers of a pre-trained neural network. The technique utilizes the semantic information latent in the deep hierarchy to aggregate on-the-fly an effective importance map. We have shown the strength of high-level image analysis versus commonly used low-level feature analysis. In addition, our technique is based on an optimization procedure that reconstructs the image from its deep features, which tends to produce much less visible artifacts.

In this work, we use a specific available pre-trained network. However, in the future we would like to consider pre-training a network with a special-purpose target in mind, so its deep features will be more relevant to this specific task. Another avenue for future work is to leverage optimization of the target image to synthesize new content. This will possibly be effective in upscaling an image into a very different aspect ratio.

Electronic Supplementary Material Supplementary material is available in the online version of this article at `https://doi.org/10.1007/s41095-021-0216-x`.

References

[1] Avidan, S.; Shamir, A. Seam carving for content-aware image resizing. *ACM Transactions on Graphics* Vol. 26, No. 3, 10–es, 2007.

[2] Wolf, L.; Guttmann, M.; Cohen-Or, D. Non-homogeneous content-driven video-retargeting. In: Proceedings of the IEEE 11th International Conference on Computer Vision, 1–6, 2007.

[3] Rubinstein, M.; Shamir, A.; Avidan, S. Improved seam

carving for video retargeting. *ACM Transactions on Graphics* Vol. 27, No. 3, Article No. 16, 2008.

[4] Wang, Y. S.; Tai, C. L.; Sorkine, O.; Lee, T. Y. Optimized scale-and-stretch for image resizing. In: Proceedings of the ACM SIGGRAPH Asia Papers, Article No. 118, 2008.

[5] Pritch, Y.; Kav-Venaki, E.; Peleg, S. Shift-map image editing. In: Proceedings of the IEEE 12th International Conference on Computer Vision, 151–158, 2009.

[6] Guo, Y. W.; Liu, F.; Shi, J.; Zhou, Z. H.; Gleicher, M. Image retargeting using mesh parametrization. *IEEE Transactions on Multimedia* Vol. 11, No. 5, 856–867, 2009.

[7] Krähenbühl, P.; Lang, M.; Hornung, A.; Gross, M. A system for retargeting of streaming video. *ACM Transactions on Graphics* Vol. 28, No. 5, https://doi.org/10.1145/1618452.1618472, 2009.

[8] Rubinstein, M.; Shamir, A.; Avidan, S. Multi-operator media retargeting. In: Proceedings of the ACM SIGGRAPH Papers, Article No. 23, 2009.

[9] Wu, H. S.; Wang, Y. S.; Feng, K. C.; Wong, T. T.; Lee, T. Y.; Heng, P. A. Resizing by symmetry-summarization. In: Proceedings of the ACM SIGGRAPH Asia Papers, Article No. 159, 2010.

[10] Panozzo, D.; Weber, O.; Sorkine, O. Robust image retargeting via axis-aligned deformation. *Computer Graphics Forum* Vol. 31, No. 2pt1, 229–236, 2012.

[11] Cho, D.; Park, J.; Oh, T. H.; Tai, Y. W.; Kweon, I. S. Weakly- and self-supervised learning for content-aware deep image retargeting. In: Proceedings of the IEEE International Conference on Computer Vision, 4568–4577, 2017.

[12] Shocher, A.; Bagon, S.; Isola, P.; Irani, M. InGAN: Capturing and retargeting the "DNA" of a natural image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4491–4500, 2019.

[13] Shaham, T. R.; Dekel, T.; Michaeli, T. SinGAN: Learning a generative model from a single natural image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4569–4579, 2019.

[14] Blau, Y.; Michaeli, T. The perception-distortion tradeoff. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6228–6237, 2018.

[15] Kiess, J.; Kopf, S.; Guthier, B.; Effelsberg, W. A survey on content-aware image and video retargeting. *ACM Transactions on Multimedia Computing, Communications, and Applications* Vol. 14, No. 3, Article No. 76, 2018.

[16] Vaquero, D.; Turk, M.; Pulli, K.; Tico, M.; Gelfand, N. A survey of image retargeting techniques. In: Proceedings of the Applications of Digital Image Processing XXXIII, Vol. 7798, 328–342, 2010.

[17] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* Vol. 60, No. 6, 84–90, 2017.

[18] Simonyan K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556, 2014.

[19] Szegedy, C.; Liu, W.; Jia, Y. Q.; 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, 1–9, 2015.

[20] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.

[21] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580–587, 2014.

[22] Girshick, R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 1440–1448, 2015.

[23] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8691.* Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 346–361, 2014.

[24] Ren, S. Q.; He, K. M.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 39, No. 6, 1137–1149, 2017.

[25] Liu, Z. G.; Wang, Z. P.; Zhang, L. M.; Shah, R. R.; Xia, Y. J.; Yang, Y.; Li, X. FastShrinkage: Perceptually-aware retargeting toward mobile platforms. In: Proceedings of the 25th ACM International Conference on Multimedia, 501–509, 2017.

[26] Song, Y.; Tang, F.; Dong, W. M.; Zhang, X. P.; Deussen, O.; Lee, T. Y. Photo squarization by deep multi-operator retargeting. In: Proceedings of the 26th ACM international Conference on Multimedia, 1047–1055, 2018.

[27] Kajiura, N.; Kosugi, S.; Wang, X. T.; Yamasaki, T. Self-play reinforcement learning for fast image retargeting. In: Proceedings of the 28th ACM International Conference on Multimedia, 1755–1763, 2020.

[28] Esmaeili, S. A.; Singh, B.; Davis, L. S. Fast-at: Fast automatic thumbnail generation using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4178–4186, 2017.

[29] Lin, J. X.; Zhou, T. K.; Chen, Z. B. DeepIR: A deep semantics driven framework for image retargeting. In: Proceedings of the IEEE International Conference on Multimedia & Expo Workshops, 54–59, 2019.

[30] Liao, J.; Yao, Y.; Yuan, L.; Hua, G.; Kang, S. B. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 120, 2017.

[31] Barnes, C.; Shechtman, E.; Finkelstein, A.; Goldman, D. B. PatchMatch: A randomized correspondence algorithm for structural image editing. In: Proceedings of the ACM SIGGRAPH 2009 Papers, Article No. 24, 2009.

[32] Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 2, 2017–2025, 2015.

[33] 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, 2414–2423, 2016.

[34] Kingma, D. P.; Ba, J. Adam: A method forstochastic optimization. *arXiv preprint* arXiv:1412.6980, 2014.

[35] Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and checkerboard artifacts. *Distill*, 2016. Available at https://distill.pub/2016/deconv-checkerboard/.

[36] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* Vol. 115, No. 3, 211–252, 2015.

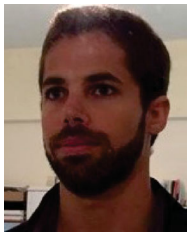
[37] Everingham, M.; van Gool, L.; Williams, C. K. I.; Winn, J.; Zisserman, A. The PASCAL visual object classes challenge 2007 (VOC2007) results. 2007. Available at http://www.pascalnetwork.org/challenges/VOC/voc2007/workshop/index.html.

[38] Rubinstein, M.; Gutierrez, D.; Sorkine, O.; Shamir, A. A comparative study of image retargeting. In: Proceedings of the ACM SIGGRAPH Asia Papers, Article No. 160, 2010.

[39] 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: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740–755, 2014.

[40] Ulyanov, D.; Lebedev, V.; Vedaldi, A.; Lempitsky, V. Texture networks: Feed-forward synthesis of textures and stylized images. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, Vol. 48, 1349–1357, 2016.

[41] Ulyanov, D.; Vedaldi, A.; Lempitsky, V. S. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint* arXiv:1607.08022, 2016.



Dov Danon is a Ph.D. student at the School of Computer Science, Tel-Aviv University. He received his B.Sc. (summa cum laude) degree in computer science and mathematics from the Ben Gurion of the Negev in 2007 and his M.Sc. degree in computer science from Tel-Aviv University in 2016. His research interests include machine learning and, in particular, unsupervised learning in image processing.

Moab Arar is a Ph.D. candidate at the School of Computer Science, Tel-Aviv University. He received his B.Sc. degree in computer engineering from the Technion Israel Institute of Technology in 2015, and his M.Sc. degree in computer science from Tel-Aviv University in 2019. His research interests span computer graphics and computer vision, with a particular focus on deep learning and machine learning methodologies for vision and rendering tasks.

Daniel Cohen-Or is a professor at the School of Computer Science, Tel-Aviv University. He received his B.Sc. (cum laude) degree in mathematics and computer science and his M.Sc. (cum laude) degree in computer science, both from Ben-Gurion University, in 1985 and 1986, respectively. He received his Ph.D. from the Department of Computer Science at the State University of New York at Stony Brook in 1991. He received the 2005 Eurographics Outstanding Technical Contributions Award. In 2015, he was named a Thomson Reuters Highly Cited Researcher. Currently, his main interests are in image synthesis, analysis and reconstruction, motion and transformations, shapes and surfaces.

Ariel Shamir is the Dean of the Efi Arazi School of Computer Science at the Interdisciplinary Center in Israel. He received his Ph.D. degree in computer science in 2000 from the Hebrew University in Jerusalem, and spent two years as a postdoctoral researcher in the computational visualisation centre at the University of Texas in Austin. He is currently an associate editor for *ACM Transactions on Graphics, Graphical Models*, and *Computational Visual Media*, and was an associate editor for *Computers and Graphics* journal (2010–2014), and *IEEE Transactions on Visualization and Computer Graphics* (2015–2017). He has also served on the program committee of many leading international conferences, including SIGGRAPH, SIGGRAPH Asia, and Eurographics. Prof. Shamir was named one of the most highly cited researchers on the Thomson Reuters list in 2015. He has a broad commercial experience of consulting for various companies including Disney Research, Mitsubishi Electric, PrimeSense (now Apple), and Verisk. He specializes in geometric modeling, computer graphics, image processing, and machine learning.