Computational Visual Media
https://doi.org/10.1007/s41095-019-0147-y
Research Article

Unsupervised natural image patch learning

Dov Danon[1] (✉), **Hadar Averbuch-Elor**[1], **Ohad Fried**[2], **and Daniel Cohen-Or**[1]

Abstract A metric for natural image patches is an important tool for analyzing images. An efficient means of learning one is to train a deep network to map an image patch to a vector space, in which the Euclidean distance reflects patch similarity. Previous attempts learned such an embedding in a supervised manner, requiring the availability of many annotated images. In this paper, we present an unsupervised embedding of natural image patches, avoiding the need for annotated images. The key idea is that the similarity of two patches can be learned from the prevalence of their spatial proximity in natural images. Clearly, relying on this simple principle, many spatially nearby pairs are outliers. However, as we show, these outliers do not harm the convergence of the metric learning. We show that our unsupervised embedding approach is more effective than a supervised one or one that uses deep patch representations. Moreover, we show that it naturally lends itself to an efficient self-supervised domain adaptation technique onto a target domain that contains a common foreground object.

Keywords unsupervised learning; metric learning

1 Introduction

Humans can easily understand what they see in different regions of an image, or tell whether two regions are similar or not. However, despite recent progress, such forms of image understanding remain extremely challenging. One way to address image understanding takes inspiration from the ability of human observers to understand image contents, even when viewing through a small observation window. Image understanding can be formalized as the ability to encode contents of small image patches into representation vectors. To keep such encoding generic, they are not predetermined by certain classes, but instead aim to project image patches into an embedding space, where Euclidean distances correlate with general similarity among image patches. As natural patches form a low dimensional manifold in the space of patches [1, 2], such an embedding of image patches allows various image understanding and segmentation tasks. For example, semantic segmentation is reduced to a simple clustering technique based on l_2 distances.

The key insight of our work is that such an embedding of image patches can be trained by a neural network in an *unsupervised* manner. Using semantic annotations allows a direct sampling of positive and negative pairs of patches that can be embedded using a triplet loss [3]. However, data labeling is laborious and expensive. Therefore, only a tiny fraction of the images available online can be utilized by supervised techniques, necessarily limiting the learning to a bounded extent. An unsupervised embedding can also be based on deep patch representations that are learned indirectly by the network, e.g., Ref. [4]. However, as we show, explicitly training the network for an embedding can achieve significantly higher performance.

In this work, we introduce an unsupervised patch embedding method, which analyses natural image patches to define a mapping from a patch to a vector, such that the Euclidean distance between two vectors reflects their perceptual similarity. We observe that the similarity of two patches in natural images is correlated with their spatial distance. In other words, patches of coherent or semantically similar segments tend to be spatially close, hence forming a surprisingly

1 Tel-Aviv University, Tel Aviv 6997801, Israel. E-mail: D. Danon, dov84d@gmail.com (✉); H. Averbuch-Elor, hadar.a.elor@gmail.com; D. Cohen-Or, dcor@tau.ac.il.
2 Stanford University, Stanford, CA 94305, USA. E-mail: ohad@stanford.edu.
Manuscript received: 2019-04-23; accepted: 2019-05-18

simple but strong correlation between patch similarity and spatial distance. Clearly, not all neighboring patches are similar (see Fig. 2). However, as we shall show, these dissimilar close patches are rare enough and uncorrelated, resulting in insignificant noise in the learning system which does not prohibit the learning.

Our embedding yields *deep images*, as each patch is mapped to a vector of 128D by a deep network. See the visualization of the deep images in the second and fourth rows of Fig. 1, obtained by projecting the 128D vectors onto their three principle directions, producing pseudo-RGB images where similar colors correspond to similar embedded points. Using our embedding technique, we further present a domain specialization method. Given a new domain that contains a common foreground object, using self-supervision, we refine the initial embedding results for the specific domain to yield a more accurate embedding.

We use a convolutional neural network (CNN) to learn a 128D embedding space. We train the network on 2.5 million natural patches with a triplet-loss objective function. Section 3 explains our embedding framework in detail. Section 4 describes our domain adaptation technique to a target domain that contains a common foreground object. In Section 5, we show that the patch embedding space learned using our method is more effective than embedding spaces that were learned with supervision or those based on hand-crafted features or deep patch representations. We further show that by fine-tuning the network to a specific domain using self-supervision, we can further increase performance.

2 Related work

Our work is closely related to dimensionality reduction and embedding techniques, image patch representation, transfer learning, and neural network based optimization. In the following we highlight directly relevant research.

Image patches can be treated as a collection of partial objects with different textures. Julesz [5] introduced textons as a way to represent texture via second order statistics of small patches. Various filter banks can be used for texture representation [6], e.g., Gabor filters [7]. Also, hierarchical filter responses have been used with great success for texture synthesis [8, 9]. All these filters are fixed and not learned from data. In contrast, we learn the embedding by analyzing the distribution of all natural patches, thus avoiding the bias of hand-crafted features.

The idea of representing a patch by its pixel values (without attempting dimensionality reduction) has had success in various applications [10]; see Barnes and Zhang [11] for a survey. In Section 5, we compare our method against a raw pixel descriptor.

Žbontar and LeCun [12] train a CNN to do stereo matching on image patches. Simo-Serra et al. [13] learn SIFT-like descriptors using a Siamese network. Both of these methods focus on invariance to viewpoint changes, whereas we aim to learn invariance to fluctuations in patch appearance of similar objects.

PatchNet [14] introduces a compact and hierarchical representation of image regions. It uses raw $L^*a^*b^*$ pixel values to represent patches. PatchTable [15] proposes an efficient approximate nearest neighbor (ANN) implementation. ANN is

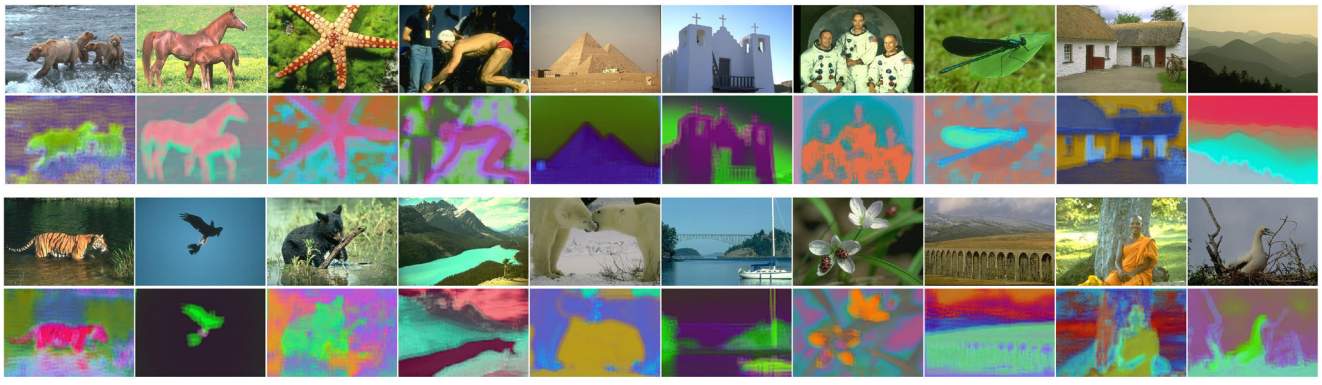


Fig. 1 Given a natural input image, our technique learns a high-dimensional embedding space, where Euclidean distances between embedded image patches reflect their similarity (visualized in pseudo-RGB colors).

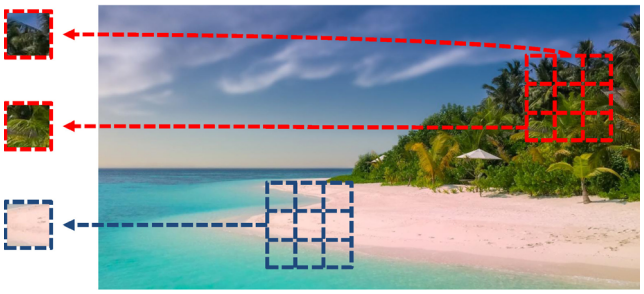


Fig. 2 Learning patch similarity from spatial distances. Our premise is that two patches sampled from the same swatch (colored in red) are more likely to be similar to each other than to a patch sampled from a distant one (colored in blue).

an orthogonal and complementary task to patch representation.

Recently, deep networks were used for image region representation and segmentation. Cimpoi et al. [16] use the last convolution layer of a convolutional neural network (CNN) as an image region descriptor. It is not suitable for patch representation, as it produces a 65k-dimensional vector *per patch.* Fully convolutional networks (FCNs) [17] prove potent for, e.g., image segmentation. We compare to FCNs in Section 5.

Our work is based on Patch2Vec [3], which also uses deep networks to train a meaningful patch representation. However, in contrary to our method, Patch2Vec is a *supervised* method that requires an annotated segmentation dataset for training.

The ideas of using spatial proximity in image space and temporal proximity for videos have been utilized in the past. For self-supervised learning, Isola et al. [18] utilize space and time co-occurrences to learn patch, frame, and photo affinities. Wang and Gupta [19] track objects in videos to generate data, also in a self-supervised manner. Wang et al. [33] introduce image extrapolation using graph matching, and exploit similarity in the spatial domain. Closer to our method, Doersch et al. [4] train a network (UVRL) to predict the spatial relationships between pairs of patches, and use the patch representation to group similar visual concepts. Pathak et al. [20] train a network to predict missing content based on its spatial surrounding. These methods learn the patch representation while training the network for a different task, and the embedding is provided implicitly. In our work, the network is directly trained for patch embedding. We compare our method against UVRL in Section 5.

Given a labeled set in a *source* domain and an unlabeled set of samples in a *target* domain, domain adaptation aims to generalize the classifier learned on the source domain to the target domain [21, 22]. It has become common practice to pre-train a classifier on a large labeled image database, such as ImageNet [23], and transfer the parameters to a target domain [24, 25]. See Patel et al. [26] for a survey of recent visual techniques. In our work, we refine our embeddings from the natural image source domain to a target domain that contains a common object. Unlike recent unsupervised domain adaptation techniques [27, 28], in our case neither domain contains labeled data.

3 Patch space embedding

In this work, we take advantage of the fact that there is a strong coherence in the appearance of semantic segments in natural images. It is expected then that nearby patches have similar appearance. The correlation between spatial proximity and appearance similarity is learned and encoded in a patch space, where the Euclidean distance between two patches reflects their appearance similarity.

The embedding patch space is learned by training a neural network using a triplet loss:

$$L(p_{\mathrm{c}}, p_{\mathrm{n}}, p_{\mathrm{f}}) = \max(0, \| f(p_{\mathrm{c}}) - f(p_{\mathrm{n}}) \|_2^2 - \| f(p_{\mathrm{c}}) - f(p_{\mathrm{f}}) \|_2^2 + m) \quad (1)$$

where $p_{\mathrm{c}}, p_{\mathrm{n}}, p_{\mathrm{f}}$ are three patches of size $s \times s$, selected from a collection of natural images, such that p_{c} is the current patch, p_{n} is a nearby patch, p_{f} is a distant patch, and m is a margin value (set empirically to 0.2). We use $s = 16$ for all our results.

To train our network, we utilize a large number of natural images (5000 images from the MIT-Adobe FiveK Dataset, in our implementation) and for each image we sample six disjoint regions, referred to as swatches. Each swatch is a 3×3 grid of patches. When sampling, we enforce a minimal distance of 3 s between swatches. In total, we sample 6 swatches per image, which is the maximal number guaranteed to fit in all our images. A triplet is formed by randomly picking two patches from one swatch, and one from another swatch. The assumption is that the two patches taken from the same swatch are close enough, while the third is distant. In our implementation,

the distant patch is always taken from the same image. The above scheme for sampling triplets is illustrated in Fig. 2, where only two swatches are illustrated, one in red and one in blue. A triplet is formed by sampling two positive patches from the red swatch, and one negative patch from the blue one. Furthermore, we adopt the principle described in Ref. [3] that selects the "hard" examples, i.e., in each epoch, we use triplets that previously were not handled well by the network. This is expressed by the following equation:

$$\mathcal{N}(p_{\mathrm{c}}, p_{\mathrm{n}}) = \{p_{\mathrm{f}} \mid L(p_{\mathrm{c}}, p_{\mathrm{n}}, p_{\mathrm{f}}) > 0\} \qquad (2)$$

Thus, the set $\mathcal{N}$ contains distant patches that the network embedded within the margin m. The network $f(p)$ is trained to create an embedding space that includes the training triplets. Once trained, $f(p)$ can embed any given patch by feed-forwarding it through the network, yielding its 128D feature vector.

To cope with outliers, we incorporate strong regularization into the network. The embedding lies only on the unit hypersphere, which prevents overfitting. The unit hypersphere provides a structure to the embedding space that is otherwise unbounded.

The architecture of our network is similar to the one used in Ref. [3], but with the required changes for supporting 16×16 patches (the network is illustrated in Fig. 5). Note that inception layers are implemented as detailed in Szegedy et al. [29].

We train the network for 1600 epochs on NVIDIA GTX 1080. Training takes approximately 24 hours. Network convergence is shown in Fig. 3. Losses during training and testing (yellow and blue, respectively) are similar. This implies that our basic assumption holds and generalizes well. Furthermore, although learning converges, convergence is not completely stable. This may be attributed to the presence of outliers in the swatches, i.e., two patches from the same swatch but not from the same segment, or two patches from different swatches but from the same segment.

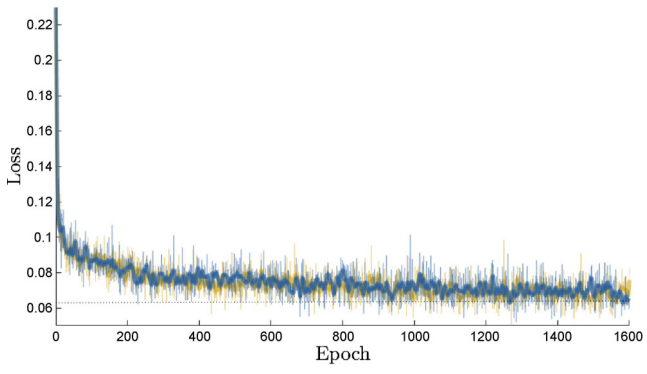


Fig. 3 Network loss convergence. The graph demonstrates the losses on the training (yellow) and test (blue) data. The loss function is not completely stable due to the presence of outlier swatches. Nonetheless, learning converges for both sets (starting from a loss of around 0.22, down to 0.07), demonstrating the network's resiliency to outliers.

We conducted several experiments that tried to distill the input to the network (the patches) using hand-crafted features. Specifically, we discarded a distant patch if its color histogram was too close to the current patch. Perhaps surprisingly, this filtering reduced accuracy by 8%. We hypothesize that this is due to the network ability to generalize better than with a hand-crafted filter as a pre-processing step.

4 Domain specialization

In Section 3, we described an unsupervised technique to encode any natural image patch as a 128D vector. Given a new domain that contains a common foreground object, we can improve the embedding by fine-tuning the network, or simply training it on patches taken from the new domain. However, we can do better, using the initial embedding obtained by the previously described method to generate a preliminary segmentation. We can then use these rough segments to "supervise" the refined embedding.

To generate the rough segments, the images are first transformed using the patch embedding so that each pixel is mapped to a vector of 128D. Next, we apply multi-region graph-cut image segmentation with 4 regions [30] (see the third row, Fig. 4). We experimented with 3–7 regions, and empirically found 4 to perform the best.

These segments are then used as supervision for fine-tuning the network, where the triplets are defined based on these segments: p_{c} and p_{n} are taken from the same foreground segment, and p_{f} is a patch taken from any other segment in the image.

In our experiments, we executed the fine-tuning process for just 400 epochs. This process improves our embedding space and makes it much more coherent (see Table 2 and Fig. 8).

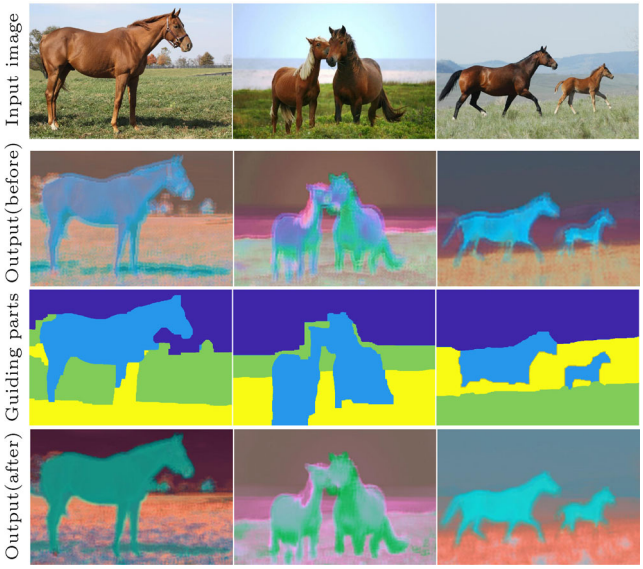


Fig. 4 Refining the embedding using self-supervision. Given a new domain that contains a common foreground object (the input images on the top), we refine our initial embedding (second row) by automatically generating semantic guiding segments (in unique colors, third row) for the training images. This yields a more coherent embedding of the common object (bottom row).

5 Results

We performed quantitative and qualitative evaluations to analyze the performance of our embedding technique. The quantitative evaluation was conducted on ground truth images from the Berkeley Segmentation Dataset (BSDS500) [31], which contains natural images that span a wide range of objects, as well as images from object-specific internet datasets of Rubinstein et al. [32]. These object-specific datasets further enabled a quantitative evaluation of our domain specialization technique.

To quantitatively demonstrate our improved performance over previous work, we adopt the measure used by Fried et al. [3]. We start by sampling "same segment" and "different segment" pairs of patches and calculate their distance in the embedding space. Next, for a given distance threshold, we predict that all pairs below the threshold are from the same segment, and evaluate the prediction (for all threshold values) by calculating the area under the receiver operating characteristic (ROC) curve.

Table 1 contains the full comparison. Notice that Ref. [3] is *supervised*, requiring an annotated segmentation dataset. The comparison to raw RGB pixels provides a more intuitive baseline. On the other hand, the accuracy of a human annotator (bottom, Table 1) demonstrates the problem ambiguity and a level of accuracy which can be considered ideal.

To qualitatively visualize the quality of our embeddings, as previously detailed, we project the 128D vectors onto their three principal directions, which enables the production of pseudo-RGB images in which similar colors correspond to similar embedded points. In Fig. 6, we visualize our embeddings and compare the results to the supervised technique of Fried et al. [3] on *their* training data. Our results are more coherent than the ones obtained with supervision, even though our method does not train on these images. In the Electronic Supplementary Material (ESM), we provide a comparison for the *full* BSDS500 dataset. Please refer to these results which demonstrate the high quality of our results.

In Fig. 7, we compare to the results of Doersch et al. [4], where the patch representation can also be obtained without supervision. For comparison purposes, we use both their pre-trained weights and

Table 1 Patch embedding evaluation. We compare our method to alternative patch representations. We report the AUC scores using l_2 distance between patch representation as means to predict whether a pair of patches comes from the same segment or not

Method	Accuracy	Unsupervised
Raw pixels (RGB)	0.69	✓
UVRL [4]	0.70	✓
Patch2Vec [3]	0.76	—
Ours	0.78	✓
Human	0.86	—

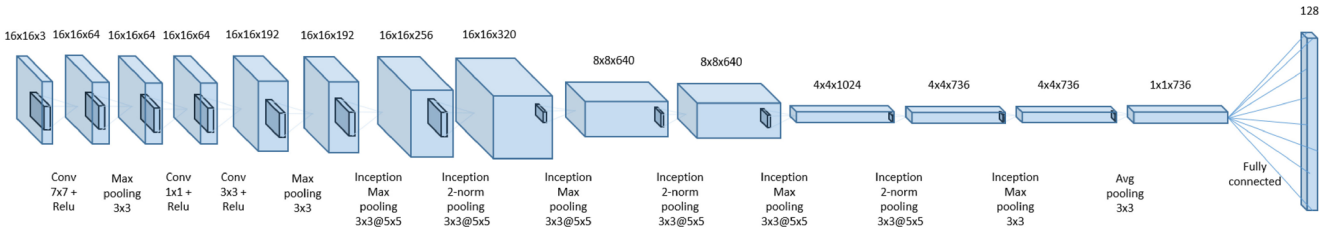


Fig. 5 Our network architecture.

Fig. 6 Supervised vs. unsupervised embedding technique. Top: input images, from the training data of Ref. [3]. Middle: results of Patch2Vec [3]. Bottom: results of our unsupervised technique. Note that although our method did not train on these images, the textures are significantly less apparent in our embeddings. This suggests that segments with similar texture are embedded to closer locations in the embedding space.

the weights retrained on BSDS500. We use their fc6 layer which performed best in our tests. Unlike our embeddings, their method does not produce similar embeddings, which are visualized by similar colors in the figure, for pixels of the same region.

To evaluate our domain specialization technique, we fine-tune our network in two ways. Firstly, we retrain the weights by simply training it on patches taken from the object-specific datasets of Rubinstein et al. [32]. The second option is described in Section 4, where we use self-supervision to refine the results in the new domain.

In Table 2, we report the AUC scores in both settings. Since the ground truth for these datasets contains only foreground–background segmentation (and not a segment for each semantic object), the AUC measure required a slight adjustment. As the background may contain many unrelated segments, we sample "same segment" pairs only from the foreground. As validated in Table 2, our method successfully learns and adjusts to the new domain. Moreover, our self-supervision scheme further boosts the performance.

Table 2 Domain specialization evaluation: AUC scores on the object-specific datasets provided by Ref. [32] before fine-tuning the network (baseline), after fine-tuning the network on patches from the dataset (fine-tuned), and after fine-tuning the network using our self-supervision technique (fine-tuned+self-supervision)

Method	Accuracy
Horses	
Baseline (before fine-tuning)	0.68
Fine-tuned (training)	0.71
Fine-tuned (testing)	0.70
Fine-tuned+self-supervision (training)	0.72
Fine-tuned+self-supervision (testing)	0.72
Airplanes	
Baseline (before fine-tuning)	0.623
Fine-tuned (training)	0.640
Fine-tuned (testing)	0.656
Fine-tuned+self-supervision (training)	0.662
Fine-tuned+self-supervision (testing)	0.672
Cars	
Baseline (before fine-tuning)	0.651
Fine-tuned (training)	0.655
Fine-tuned (testing)	0.653
Fine-tuned+self-supervision (training)	0.670
Fine-tuned+self-supervision (testing)	0.670

In Fig. 8, we qualitatively demonstrate the improvement over samples belonging to the HORSE dataset, half of them belong to the training set and the other half to the test set. Since one could not tell the samples apart, in the figure they are mixed together. As the figure illustrates, the colors, and thereby the embeddings, of the horses' parts are more compatible and in general more homogeneous. For more results, see the ESM.

6 Summary, limitations, and future work

We have presented an unsupervised patch embedding technique, where the network learns to map natural image patches into 128D codes such that the l_2 metric reflects their similarity. We showed that the triplet loss that we use to train the network explicitly for embedding outperforms other embeddings that are inferred by deep representations learned for other tasks or designed specifically to learn similarities between patches. Generally speaking, learning to embed by a network has its limitations as it is applied at the patch level. Feeding forward patches in a network is a computationally-intensive task, and analyzing an image as a series of patches is time consuming. Parallel analysis of a multitude of patches,

Fig. 7 Comparison between our embedding and one inferred by deep representations, UVRL [4], using their pre-trained weights (second row) and also by retraining them on BSDS500 (third row). As demonstrated above, our technique maps pixels from similar regions to closer values.

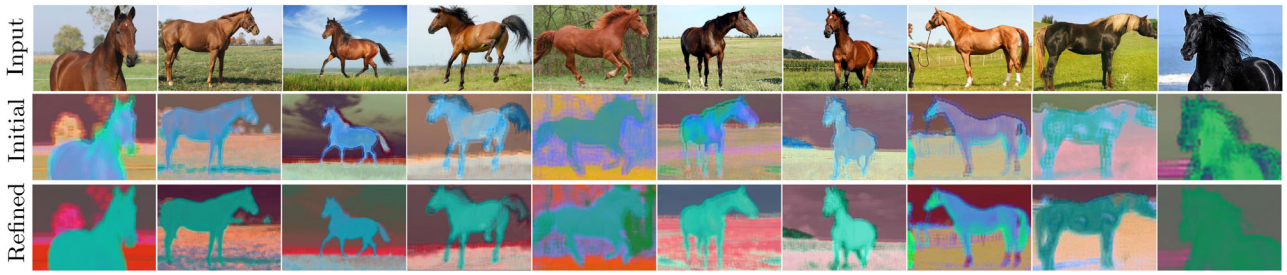


Fig. 8 Refining the embeddings to the HORSE domain. Above, we demonstrate the embeddings before and after the domain specialization stage. As shown, and quantitatively stated in Table 2, the embeddings of the objects (e.g., the horses) are more coherent after refinement.

possibly overlapping ones, can significantly accelerate the process.

To improve the performance and transfer the learning into a new domain, we utilize the embedding obtained by a trained network as self-supervision. The embedded image is segmented by a naive method, to yield a rough segmentation. As demonstrated, these segments, although imperfect, can successfully supervise the refinement of the network for the given new domain. However, we believe this can be further improved by using more advanced segmentation methods. In the future, we wish to consider conservative segmentation, where the segments may not necessarily cover the entire image, excluding regions with low confidence.

Furthermore, in future, we would like to utilize our embedding technique to advance segmentation and foreground extraction methods. In particular, we hope to analyze large sets of embedded images, aiming to co-segment the common foreground of a weakly supervised set. We believe that the common foreground object can provide self-supervision to further improve the embedding performance.

Electronic Supplementary Material Supplementary material is available in the online version of this article at `https://doi.org/10.1007/s41095-019-0147-y`.

References

[1] Matviychuk, Y.; Hughes, S. M. Exploring the manifold of image patches. In: Proceedings of Bridges, 339–342, 2015.

[2] Shi, K.; Zhu, S.-C. Mapping natural image patches by explicit and implicit manifolds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–7, 2007.

[3] Fried, O.; Avidan, S.; Cohen-Or, D. Patch2Vec: Globally consistent image patch representation. *Computer Graphics Forum* Vol. 36, No. 7, 183–194, 2017.

[4] Doersch, C.; Gupta, A.; Efros, A. A. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, 1422–1430, 2015.

[5] Julesz, B. Textons, the elements of texture perception, and their interactions. *Nature* Vol. 290, No. 5802, 91–97, 1981.

[6] Randen, T.; Husoy, J. H. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 21, No. 4, 291–310, 1999.

[7] Gabor, D. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers – Part III: Radio and Communication Engineering* Vol. 93, No. 26, 429–441, 1946.

[8] De Bonet, J. S.; Viola, P. Texture recognition using a non-parametric multi-scale statistical model. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 641–647, 1998.

[9] Heeger, D. J.; Bergen, J. R. Pyramid-based texture analysis/synthesis. In: Proceedings of the IEEE International Conference on Image Processing 648–650, 1995.

[10] Varma, M.; Zisserman, A. Texture classification: Are filter banks necessary? In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, II-691, 2003.

[11] Barnes, C.; Zhang, F. L. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media* Vol. 3, No. 1, 3–20, 2017.

[12] Žbontar, J.; LeCun, Y. Stereo matching by training a convolutional neural network to compare image patches. *The Journal of Machine Learning Research* Vol. 17, No. 1, 2287–2318, 2016.

[13] Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Fua, P.; Moreno-Noguer, F. Discriminative learning of deep convolutional feature point descriptors. In: Proceedings of the IEEE International Conference on Computer Vision, 118–126, 2015.

[14] Hu, S.-M.; Zhang, F.-L.; Wang, M.; Martin, R. R.; Wang, J. PatchNet: A patch-based image representation for interactive library-driven image editing. *ACM Transactions on Graphics* Vol. 32, No. 6, Article No. 196, 2013.

[15] Barnes, C.; Zhang, F.-L.; Lou, L.; Wu, X.; Hu, S.-M. PatchTable: Efficient patch queries for large datasets and applications. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 97, 2015.

[16] Cimpoi, M.; Maji, S.; Vedaldi, A. Deep filter banks for texture recognition and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3828–3836, 2015.

[17] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440, 2015.

[18] Isola, P.; Zoran, D.; Krishnan, D.; Adelson, E. H. Learning visual groups from co-occurrences in space and time. *arXiv preprint* arXiv:1511.06811, 2015.

[19] Wang X.; Gupta, A. Unsupervised learning of visual representations using videos. In: Proceeding of the IEEE International Conference on Computer Vision, 2794–2802, 2015.

[20] Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; Efros, A. A. Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2536–2544, 2016.

[21] Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; Vaughan, J. W. A theory of learning from different domains. *Machine Learning* Vol. 79, Nos. 1–2, 151–175, 2010.

[22] Chen, M.; Xu, Z.; Weinberger, K. Q.; Sha, F. Marginalized denoising autoencoders for domain adaptation. In: Proceedings of the 29th International Conference on Machine Learning, 1627–1634, 2012.

[23] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S. A. et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* Vol. 115, No. 3, 211–252, 2015.

[24] Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1717–1724, 2014.

[25] Razavian, A. S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 806–813, 2014.

[26] Patel, V. M.; Gopalan, R.; Li, R. N.; Chellappa, R. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine* Vol. 32, No. 3, 53–69, 2015.

[27] Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on Machine Learning, Vol. 37, 1180–1189, 2015.

[28] Kodirov, E.; Xiang, T.; Fu, Z.; Gong, S. Unsupervised domain adaptation for zero-shot learning. In: Proceedings of the IEEE International Conference on Computer Visio, 2452–2460, 2015.

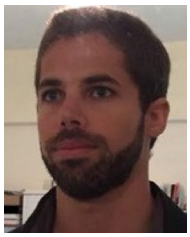
[29] Szegedy, C.; Liu, W.; Jia, Y. Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–9, 2015.

[30] Bagon, S. Matlab wrapper for graph cut. 2006. Available at http://www.wisdom.weizmann.ac.il/∼bagon/matlab.html.

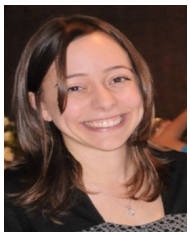
[31] Arbeláez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 33, No. 5, 898–916, 2011.

[32] Rubinstein, M.; Joulin, A.; Kopf, J.; Liu, C. Unsupervised joint object discovery and segmentation in Internet images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1939–1946, 2013.

[33] Wang, M.; Lai, Y.; Liang, Y.; Martin, R. R.; Hu, S.-M. BiggerPicture: Data-driven image extrapolation using graph matching. *ACM Transactions on Graphics* Vol. 33, No. 6, Article No. 173, 2014.



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 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.



Hadar Averbuch-Elor is a Ph.D. student at the School of Electrical Engineering, Tel-Aviv University, and a research scientist at Amazon. She received her B.Sc. (cum laude) degree in electrical engineering from the Technion in 2012. She worked as a computer vision algorithm developer in the defense industry from 2011 to 2015. Her research interests include computer vision and computer graphics, focusing on unstructured image collections and unsupervised techniques.

Ohad Fried is a postdoctoral research scholar at the School of Computer Science, Stanford University, and a fellow in the Brown Institute for Media Innovation. He received his B.Sc. (magna cum laude) degree in computer science and computational biology and M.Sc. (cum laude) degree in computer science, both from the Hebrew University, in 2010 and 2012 respectively. He received his Ph.D. degree from the Department of Computer Science at Princeton University in 2017. Currently, his main interests are visual communication methods at the intersection of graphics, vision, and HCI.

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 M.Sc. (cum laude) degree in computer science, both from Ben-Gurion University, in 1985 and 1986, respectively. He received his Ph.D. degree 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 a few areas: image synthesis, analysis and reconstruction, motion and transformations, shapes, and surfaces.

Other papers from this open access journal are available free of charge from http://www.springer.com/journal/41095. To submit a manuscript, please go to https://www.editorialmanager.com/cvmj.