Image Shape Manipulation from a Single Augmented Training Sample Supplementary Material

Yael Vinker* Eliahu Horwitz* Nir Zabari Yedid Hoshen
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel.
{yael.vinker, eliahu.horwitz, nir.zabari, yedid.hoshen}@mail.huji.ac.il

Contents

*Equal contribution

A. A study of different augmentations

In this section we analyse the effect of different augmentations under the proposed framework. We trained our method by using different combinations of various augmentation methods i.e. crop, flip, shear, rotation, cutmix based [6](i.e. randomly swapping patches within the same single training image) and TPS. Furthermore, in order to improve the robustness to manual editing of the edges, we incorporate edge augmentation in the primitive by using randomly sampled σ values for the canny edge detector (i.e. controlling the scale of the edges, larger σ result in coarser scale edges while smaller σ result finer scale edges).

We used the same "dataset" for all the experiments. The dataset follows the video-based evaluation method presented in Sec. 4.2 of the main paper. All images are of size 480×480 and the primitives are a combination of edges and segmentations (extracted using face-parsing.PyTorch). For cutmix-like augmentations we sampled patches of random size in $[32, 96]$. We shear by $s \sim U(-0.3, 0.3)$. For rotations we uniformly sample $r \sim U(-10, 10)$ degrees. The full breakdown and results are presented in Tab. 1. As can be seen from Tab. 1 and Fig. 1 TPS has a significant role in the success of our method. Additionally, we can see in that the σ augmentation improves the reconstruction of fine details such as the teeth.

crop	flip	sheer	rotation	cutmix	tps	canny σ	SIFID ↓	LPIPS ↓
✗	✓	✗	✗	✗	✗	✗	0.25	0.15
✗	✓	✗	✗	✗	0.3	✗	0.15	0.19
✗	✓	✗	✗	✗	0.6	✗	0.14	0.18
✓	✓	✗	✗	✗	0.99	✗	0.10	0.05
✗	✓	✗	✓	✗	0.8	✗	0.10	0.07
✗	✓	✗	✗	✗	0.9	✗	0.10	0.07
✗	✓	✗	✗	✓	0.99	✗	0.10	0.03
✗	✓	✓	✓	✗	0.8	✗	0.10	0.04
✗	✓	✗	✗	✗	0.99	✗	**0.09**	0.05
✗	✓	✓	✓	✗	0.9	✗	0.10	0.04
✗	✓	✓	✓	✗	0.99	✗	0.10	0.05
✗	✓	✗	✗	✗	0.99	✓	0.10	0.02
✗	✓	✓	✓	✓	0.99	✓	0.10	**0.01**

Table 1: *Types of augmentations* The "TPS" column indicates the portion for which TPS was applied.

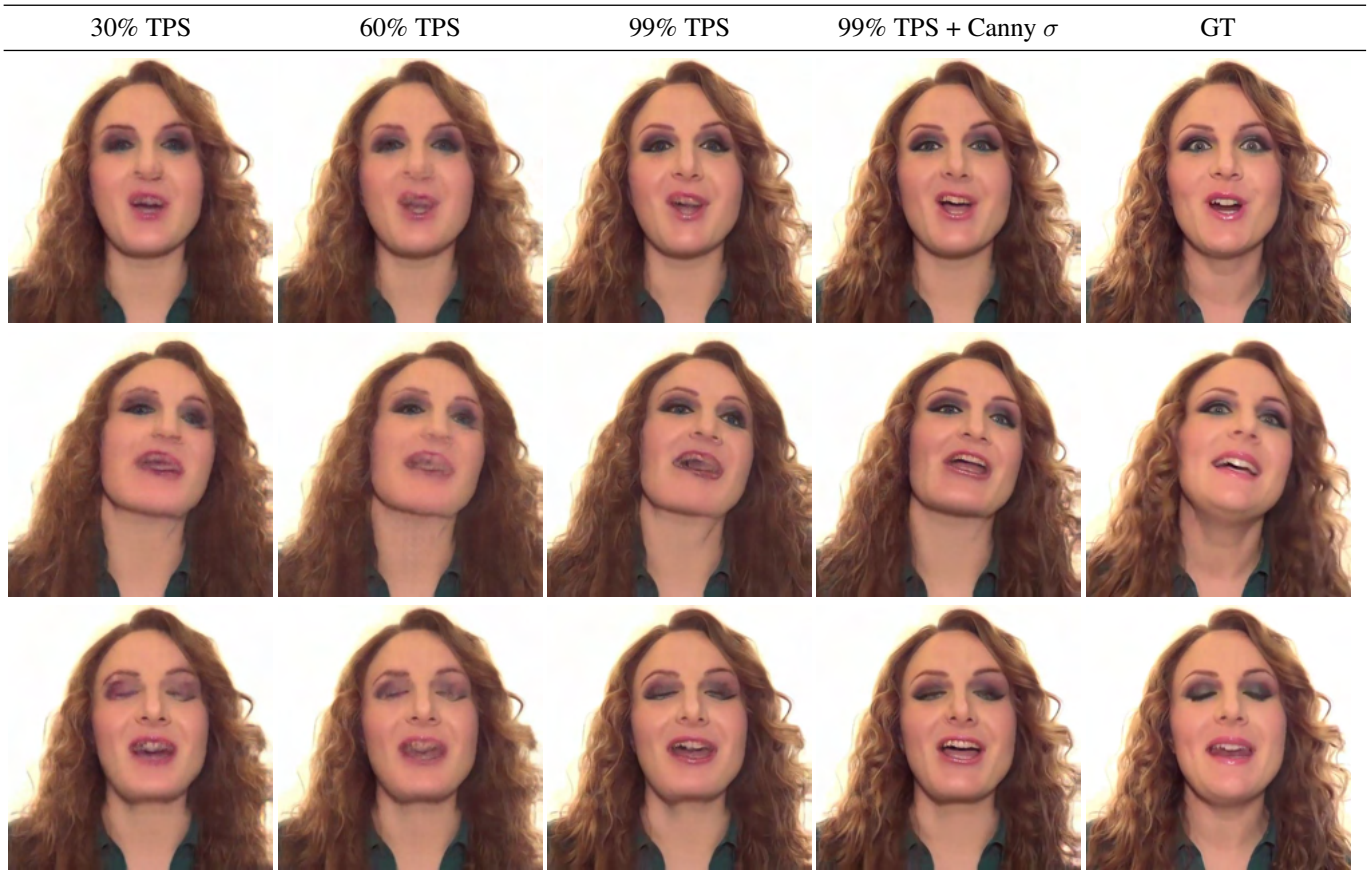


Figure 1: *Effects of augmentations*

B. Empty space interpolation

In this section we stress test our method's ability to handle regions with little guidance. In this example, the nose of the cat was shifted progressively downwards, forcing the network to interpolate the missing space. We observe the network synthesizes attractive images for moderate empty regions, however, as the empty region gets larger, the network looks for similar regions to fill the newly created void. These regions will often be areas which exhibit low amounts of detail in the primitive representation. In our case we can notice that for larger shifts, the empty space becomes greener until eventually it inpaints a background patch. We conclude that at a certain point, the network fails to learn the spatial relationship among objects in the image (i.e. that the background can not be placed on the cat's face) and satisfies the given constraint using neighboring information (as was analysed above).

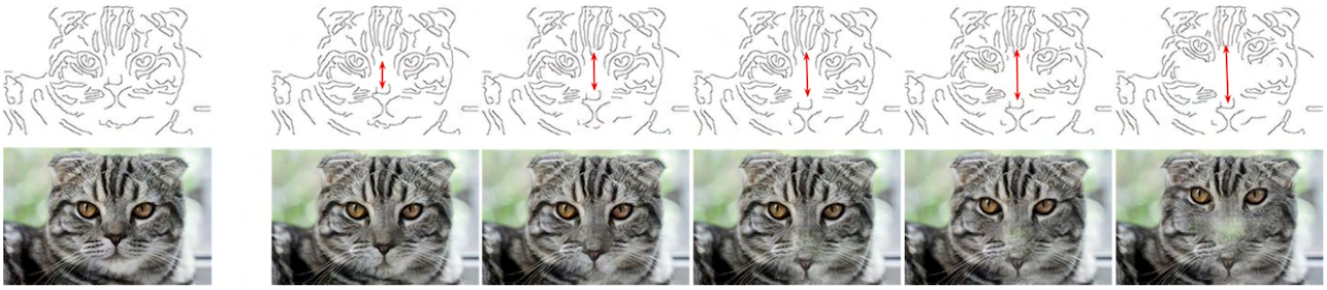


Figure 2: Evaluation of the ability of our network to interpolate across empty space regions. The two leftmost columns show the training image pair, we gradually increase the distance between the eyes and nose of the cat, and feed the test images to the network, the corresponding output of each test image is shown in the second row. Our method generates attractive interpolations for moderate changes, the performance deteriorates for larger interpolations.

C. TPS generalization improvement

Let us consider the train and test edge-image pairs presented in Fig. 3. We input each edge map through an ImageNet-trained ResNet50 network and computed the activations at the end of the third residual block. For each pixel in the activation grid of the test image, we computed the nearest-neighbor (1NN) distance to the most similar activation of the train image. We then performed 50 TPS augmentations to the training image, and repeated the 1NN computation with the training set now containing the activations of the original training image and its 50 augmentation. Let us compare the 1NN distances presented in Fig. 3 with and without TPS augmentations. Naturally, the 1NN distance decreased for the TPS-augmented training set due to its larger size. More interestingly, we can see that several face regions which prior to the augmentations did not have similar patches in the input, now have much lower distance (while more significant changes might not be possible to describe by TPS). In Fig. 3, we present the results of our method when trained on the training edge-image pair (shown in the leftmost column) and evaluated on the test edge. We can see that the prediction error (L_1 difference between ResNet50 activations of the predicted and the true test image) appears to be strongly related to the 1NN distance with TPS-augmentations. This gives some evidence to the hypothesis that the network recalls input-output pairs seen in training. It also gives an explanation for the effectiveness of TPS training, namely increasing the range of input-output pairs thus generalizing to novel images.

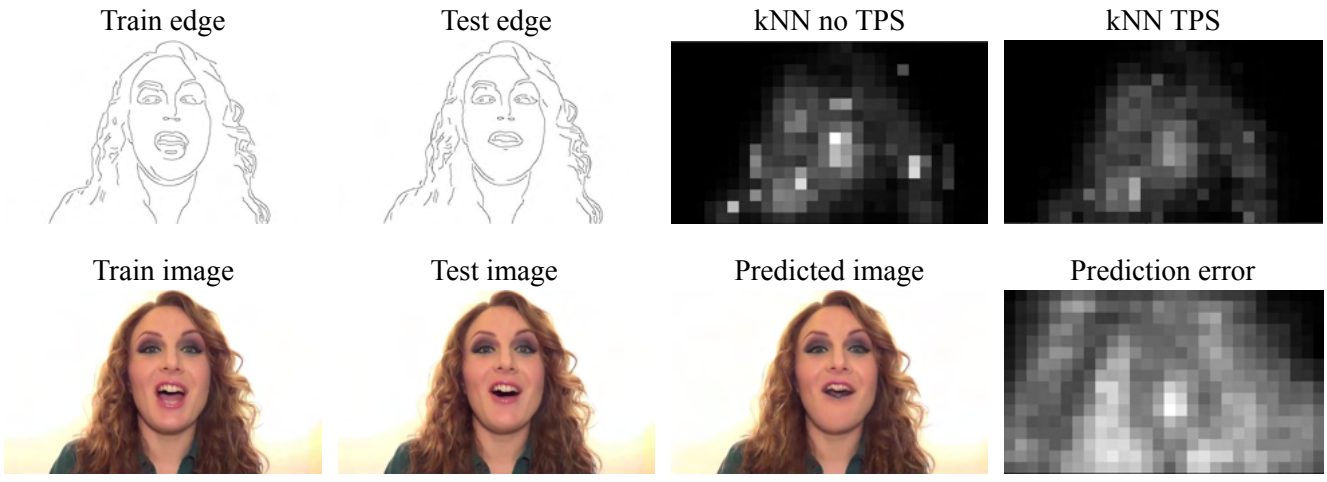
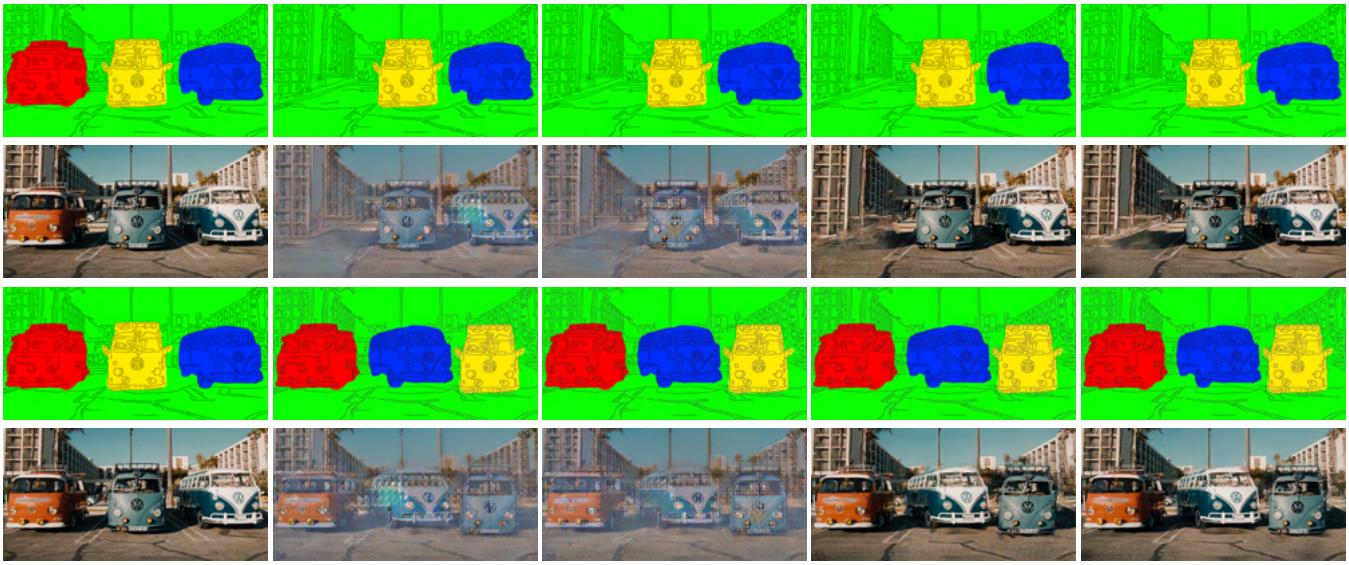


Figure 3: An analysis of the benefits of TPS. We show the kNN distance between patches in the test and train frames with and without TPS augmentations (top-right). We can see that TPS augmentation decreases the kNN distance, in some image regions the decrease is drastic suggesting the patches there can be obtained by deformations of training patches. The kNN-TPS distance appears to be correlated with the regions where the prediction error of our method is large. This analysis suggests that by artificially increasing the diversity of patches, single-image methods can generalize better to novel images.

D. An ablation of the loss objective

We compare the results of our method, DeepSIM, using the original cGAN loss as in the base Pix2PixHD architecture vs. non-adversarial losses - the simple L_1 loss and the percetual loss based on the difference of VGG activations. We can see that on this image both non-adversarial losses fail. Note that at lower resolutions non-adversarial losses do indeed succeed but do not generate results of comparable sharpness of the cGAN loss. Additionally, we performed the experiment with the cGAN but without the VGG perceptual loss, the results are presented below. It can be seen that without the VGG loss, the results are reduced in quality and contain grainy artifacts.

Training	Perceptual Loss	L1 Loss	Ours w/o VGG	Ours w/ VGG



E. Ablation of the combined primitive for the cars image

We present an ablation of the combined primitive representation (edges+segmentation) for the Cars image. In Fig. 4, we present results for a manipulation on the Cars image using edge-only, segmentation-only and combined. We can see that the combined primitive generates attractive artifact free results.
In Fig. 5 we present a qualitative comparison between different primitives on two frames from the LRS2 datasets. Although all primitives generate surprisingly good results, given the training on just a single image, the combined primitive generates cleaner outputs with fewer artifacts.

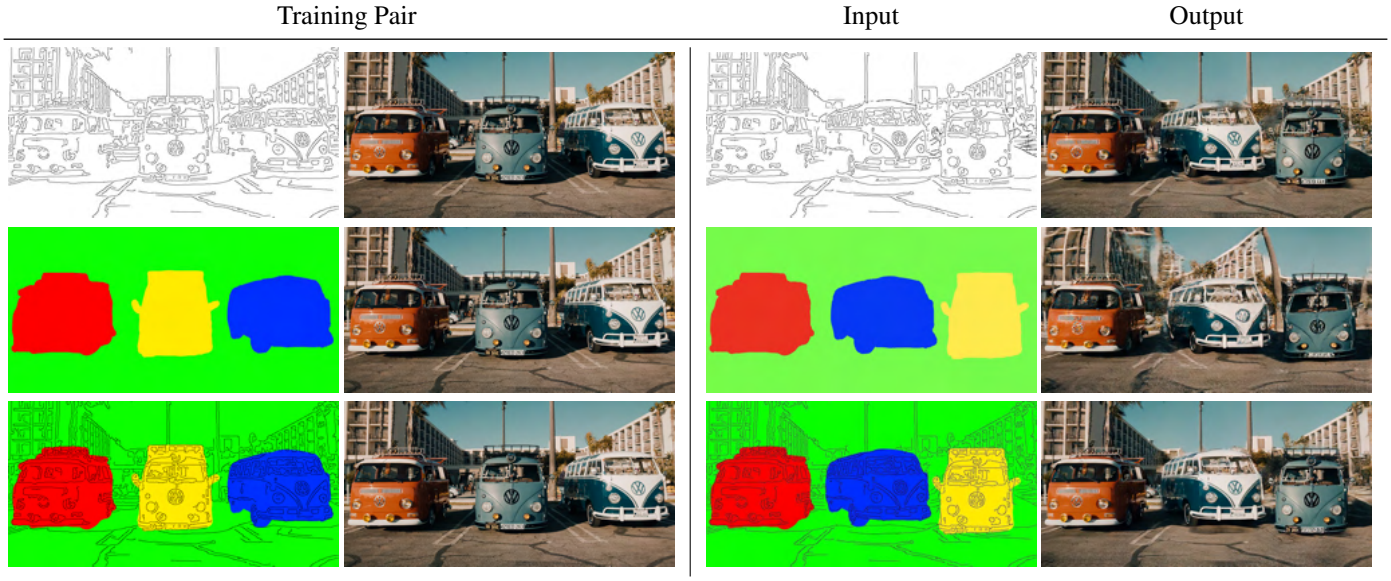


Figure 4: *Ablation of the combined primitive (edges+segmentation).* (top) edges-only (center) segmentation-only (bottom) combined. We can see that edges-only creates wrong associations between objects, segmentation-only fails to generate the fine details correctly (e.g. building), whereas the combined primitive achieves strong results.

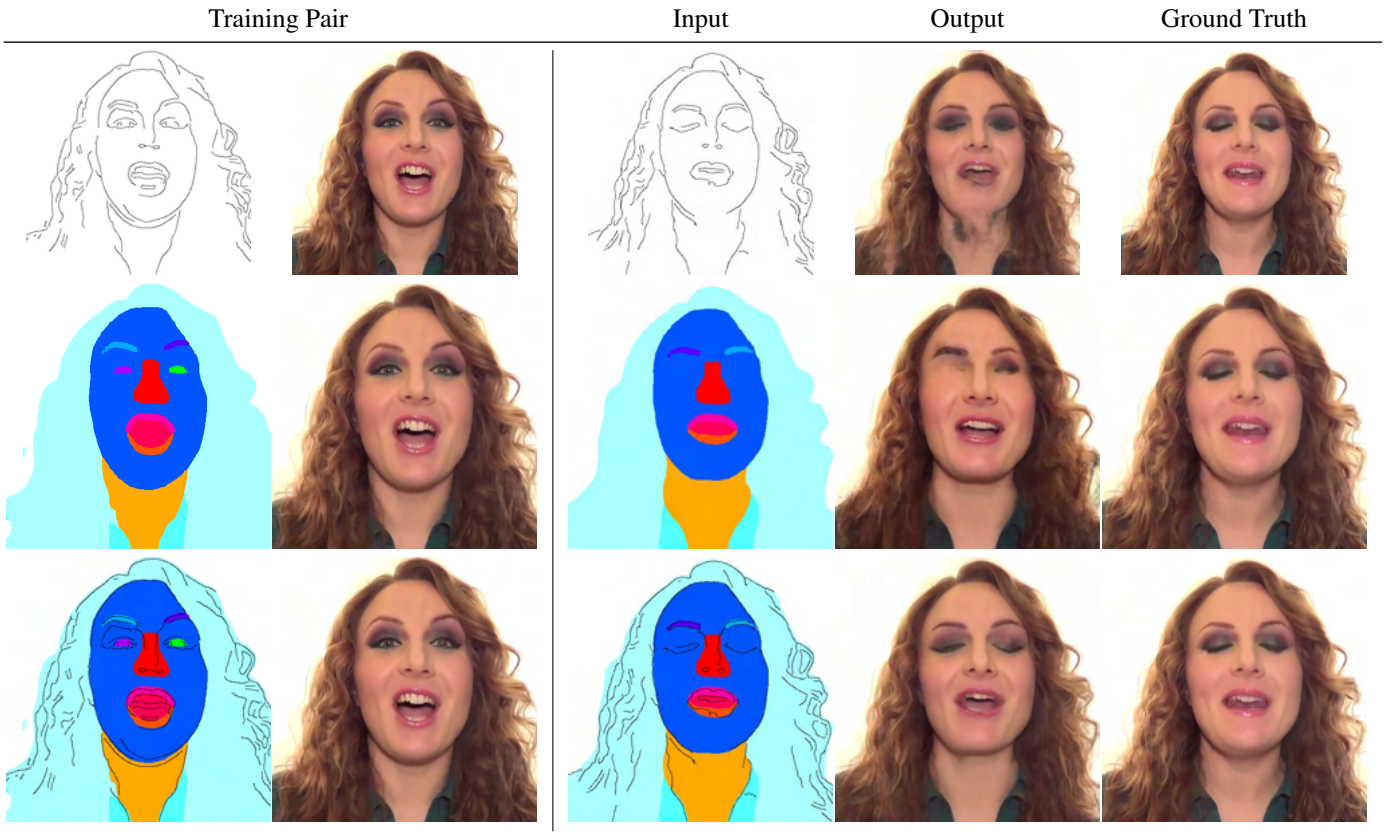


Figure 5: *Ablation of the combined primitive (edges+segmentation).* First row - The edges primitive is missing the chin. Second row - The segmentation primitive is missing the left eye. Third row - The combined primitive successfully recovered both the chin and the eye consistently with the ground truth.

F. Out-of-distribution images using pretrained model

We manually labelled the semantic and instance segmentation maps of the Cars image, and pass it to a Pix2PixHD pre-trained by the authors on the Cityscapes dataset (containing street scenes of cars, roads and buildings). We see in Fig. 6 that Pix2Pix-HD pre-trained on a large dataset does not generalize well to out-of-distribution inputs whereas our single-image method did.

Figure 6: *Out-of-Distribution results.* Running full pretrained cityscapes Pix2PixHD on a primitive representation of an out-of-distribution image. The network was not able to generalize well and generated unsatisfactory results.

G. Video Frames

A visual evaluation on a few frames from the Cityscapes dataset can be seen in Fig. 7. We compare our method to the results of Pix2PixHD-SIA, where "SIA" stands for "Single Image Augmented" e.g. a Pix2PixHD model that was trained on a single image using random crop-and-flip warps but not TPS. We can observe that our method is able to synthesize

very different scene setups from those seen in training, including different numbers and positions of people. Our method outperforms significantly in terms of fidelity and quality than Pix2PixHD-SIA indicating that our proposed TPS augmentation is critical for single image conditional generation.

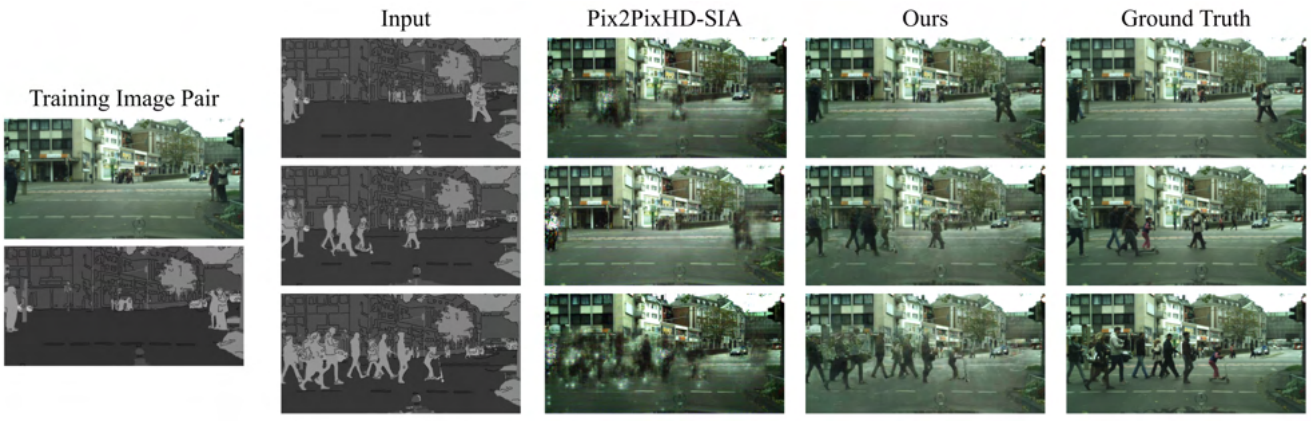


Figure 7: Several sample results from the Cityscapes dataset. We train each model on the segmentation-image pair on the left. We then use the models to predict the image, given the segmentation maps (second column from left). Our method is shown to perform very well on this task, generating novel configurations of people not seen in the training image.

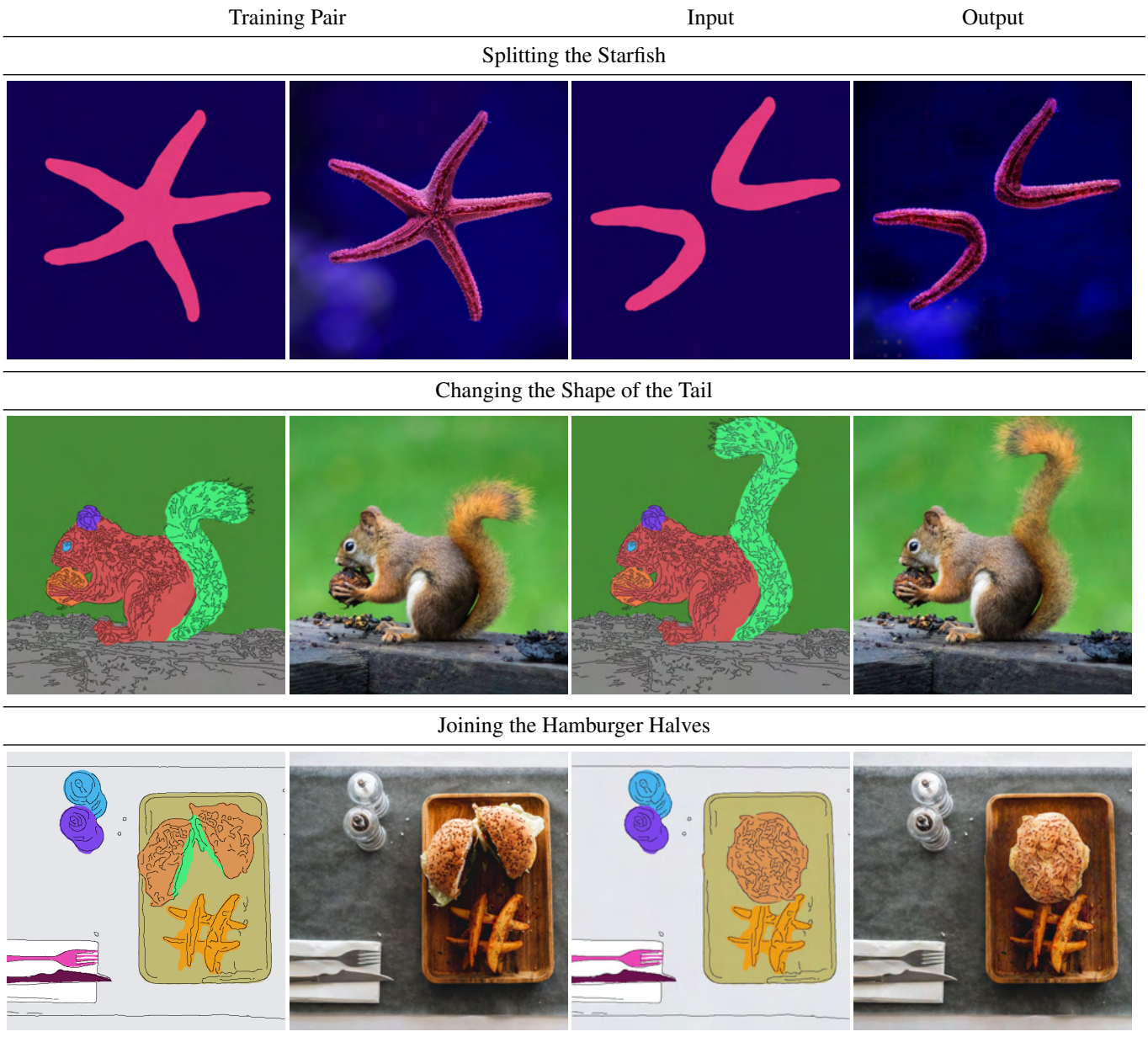
H. Qualitative comparison details

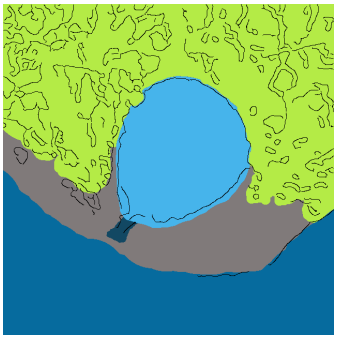
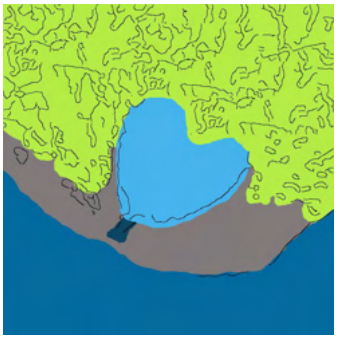
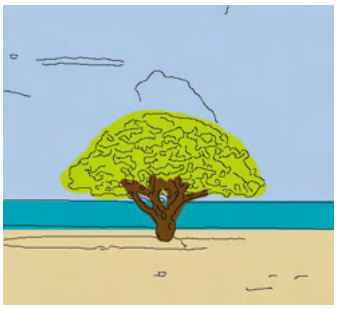
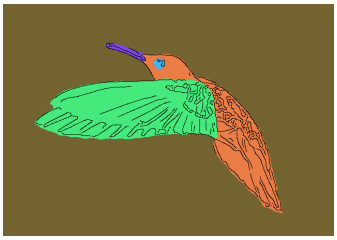
Below we provide the technical details used for our new video-based benchmark for conditional single image generation spanning a range of scenes. For the qualitative comparisons we use all 16 video segments from the Cityscapes dataset [2] provided by the code in vid2vid [5] and Few-shot-vid2vid [4]. These sequences are labelled aachen-000000 to aachen-000015 leftImg8bit. For each sequence, we train on frame 000000 and test using frames 000001 to 000029. We use the segmentation maps provided as image primitives. We also use the first 5 videos in the public release of the *Oxford-BBC Lip Reading Sentences 2* (LRS2) dataset containing videos of different speakers. We extract their edges using a Canny edge detector[1]. In total, our evaluation set contains 464 Cityscapes frames and 239 LRS2 frames.

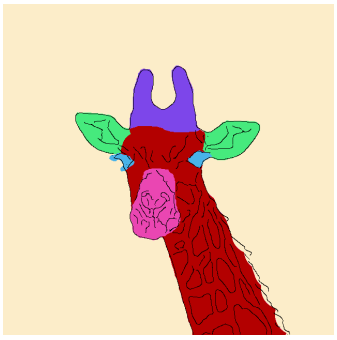
I. Additional Results

We present additional results of our method, DeepSIM, on a range of manipulations on different images. The manipulation fall into four categories: (1) Manipulations. (2) Removals. (3) Additions. (4) Single Image Animation. In (5) we provide visual comparison to Image Analogies [3], a classic method in the field of image-to-image translation.

I.1. Manipulations



Training Pair	Input	Output
Changing the Shape of the Lake		
		
Moving the Tree		
		
Making the Beak Longer		
		

Training Pair	Input	Output
Making the Body Wider		
		
Changing the Shape of the Ears of the Top Lama		
		
Changing the Shape of the Horns		
		

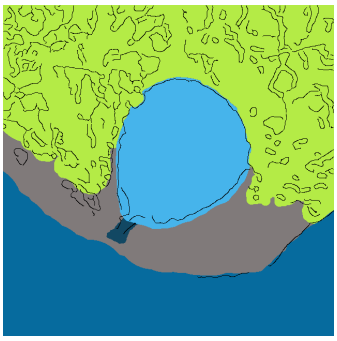
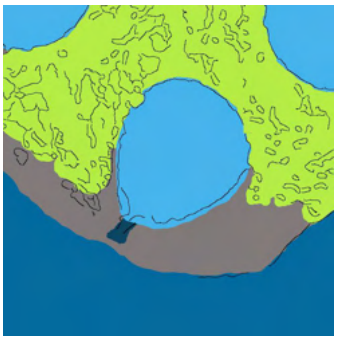
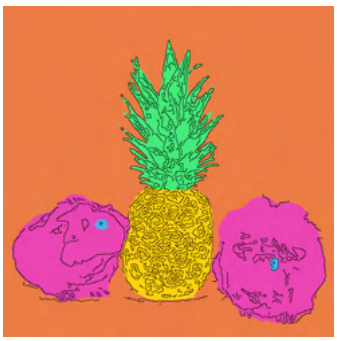
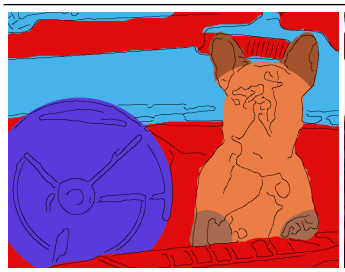
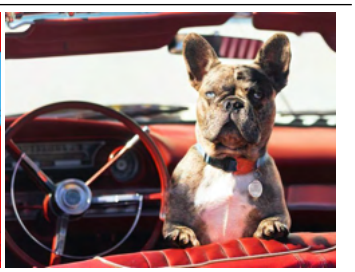
Training Pair	Input	Output
Making the Dress Longer		
 		
Changing the Shoulder Area		
 		
Changing the Shoulder Area		
 		
Changing the Cut at the Bottom		
 		

I.2. Removals

Training Pair	Input	Output
Removing the Fork		
		
Removing the Handcuffs		
		
Removing the Right Hand		
		

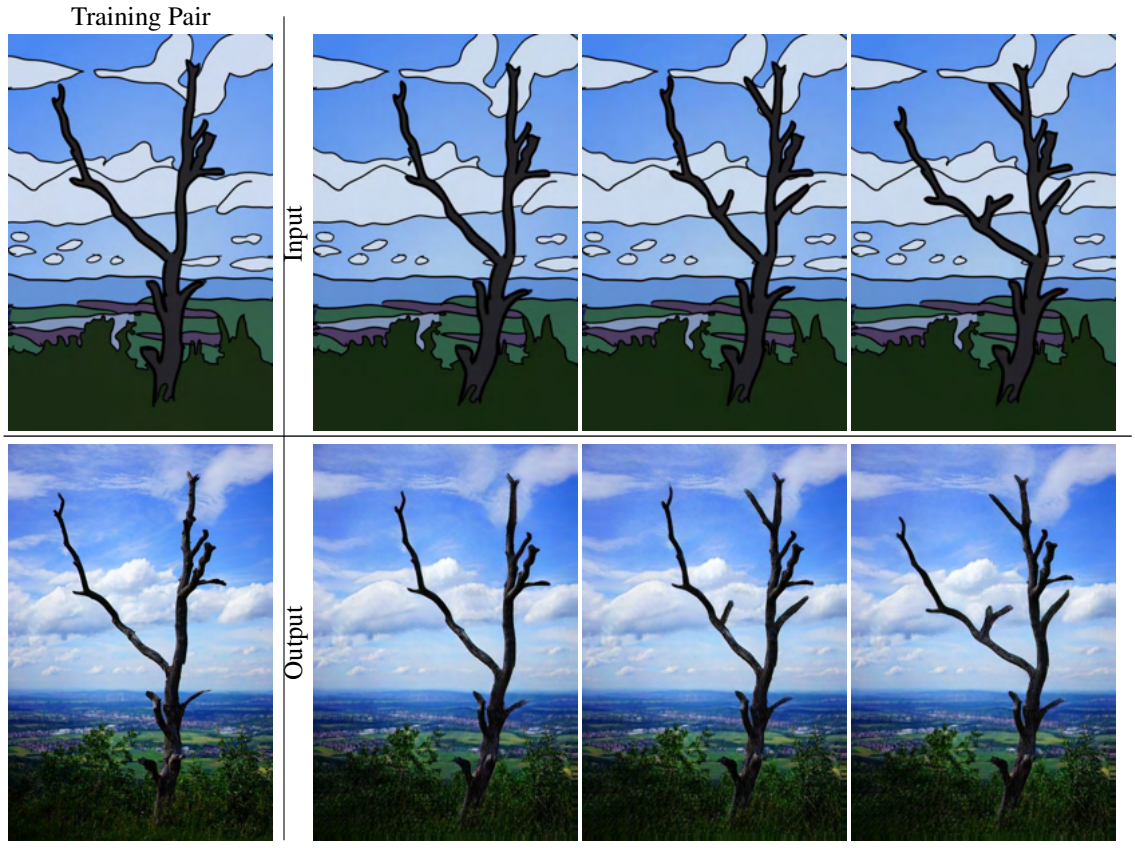
Training Pair	Input	Output
Removing Arms		
		
Removing the Teeth of the Top Lama		
		

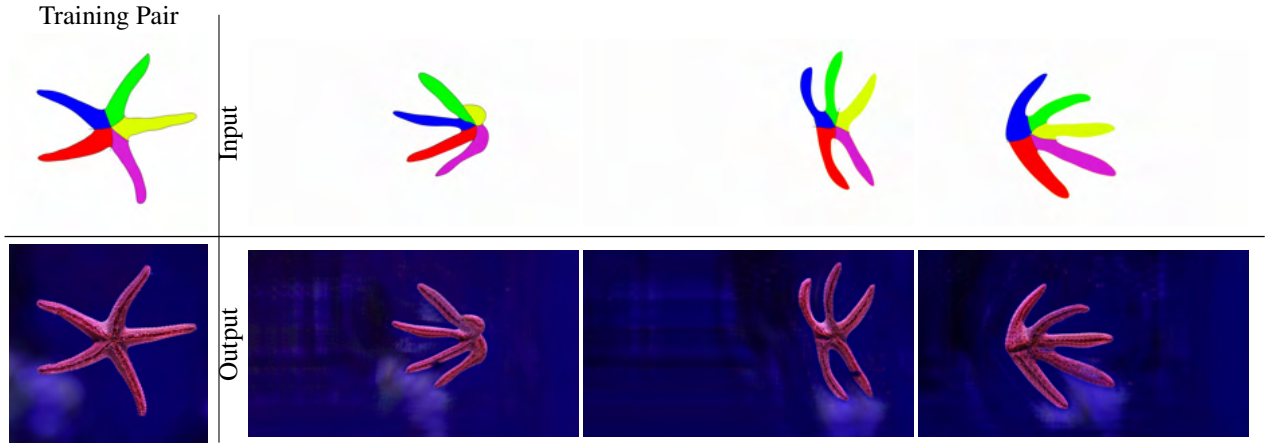
I.3. Additions

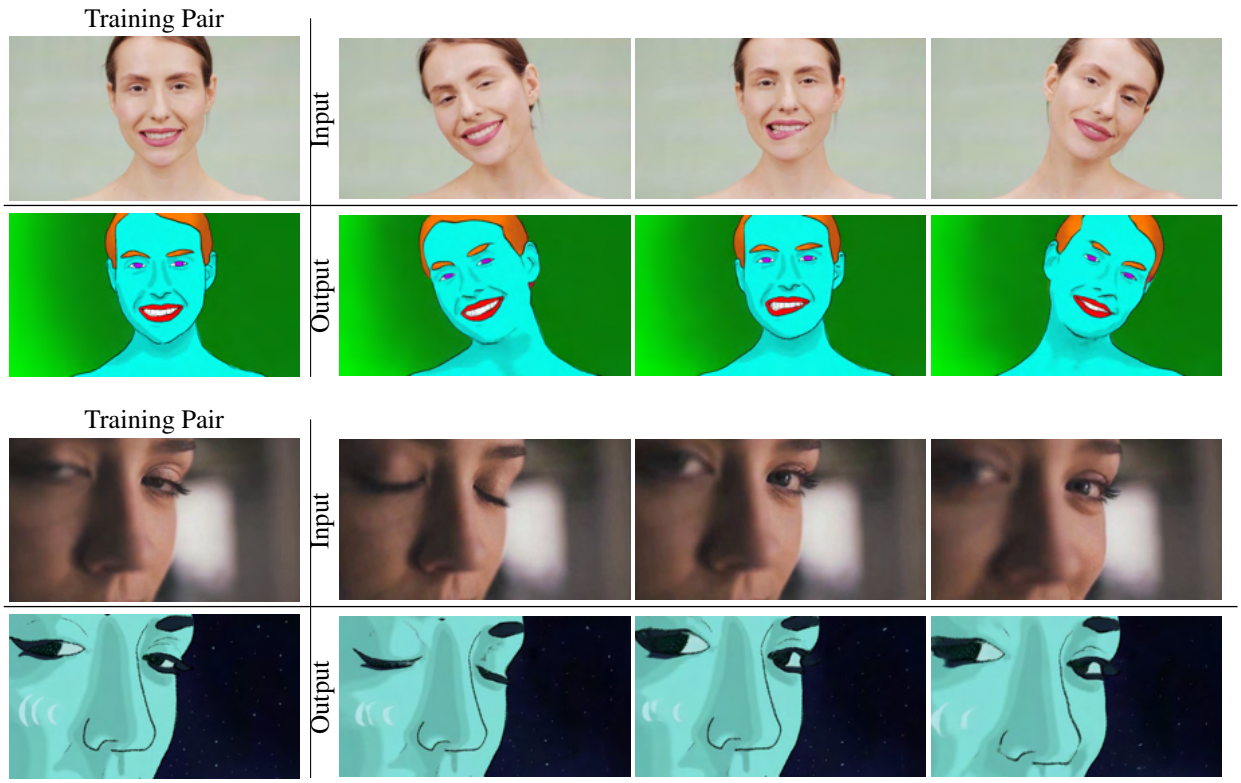
Training Pair		Input	Output
Adding More Lakes			
			
Adding Stems			
			
Adding the Left Paw			
			
Adding an Arm			
			

I.4. Single Image Animation

As described in the paper, after training we can use DeepSIM to create a short animated clip in the primitive domain, feeding it frame-by-frame to the trained model we obtain a photorealistic animated clip. In addition, DeepSIM can be used also in the opposite direction. The following figures showcase a few frames from each clip. We strongly encourage the reader to view the videos on our project page.

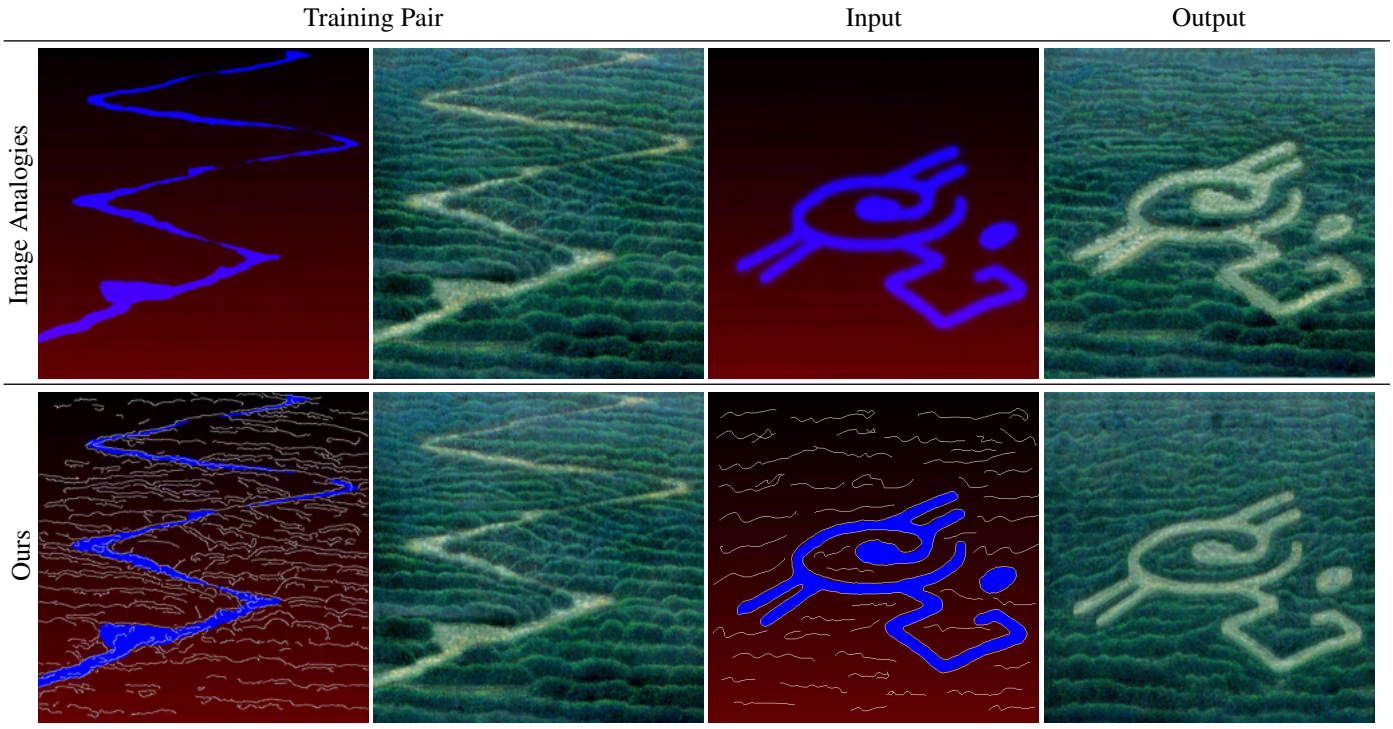





Training Pair
Input
Output
Training Pair
Input
Output

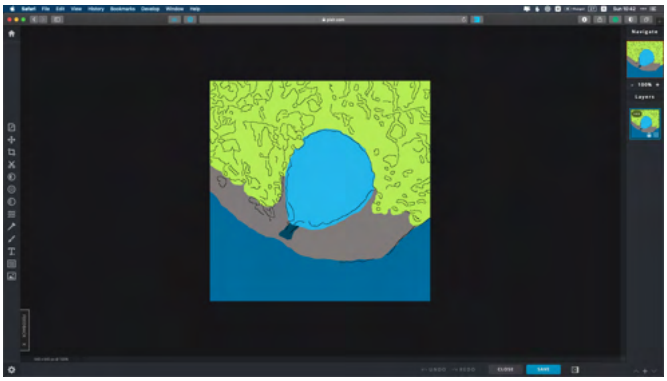
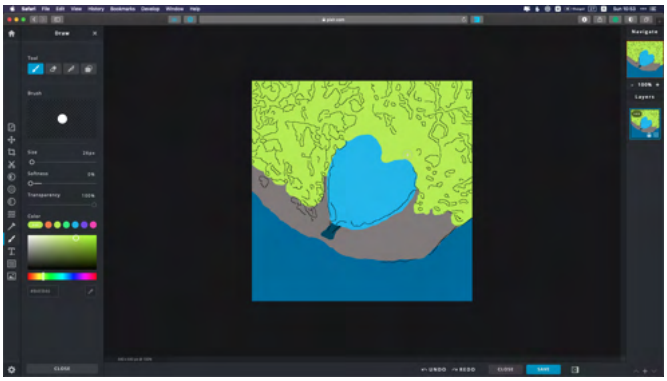
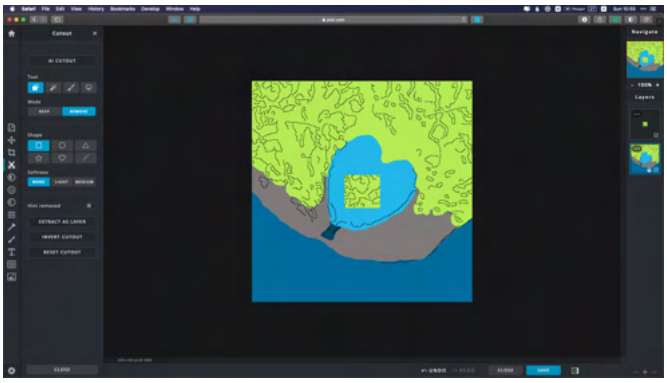
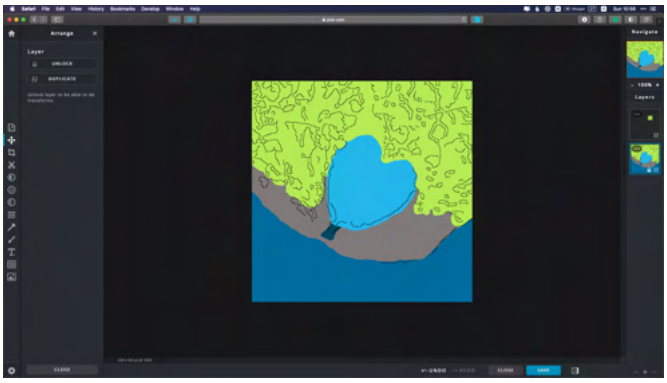
I.5. Comparison to Image Analogies

Image Analogies by Hertzmann et al. [3] is based on finding multi-scale patch-level analogies between a pair of images in order to apply a wide variety of “image filter” effects to a given image. Below is a comparison of our method to theirs using the ”path” example from the ”texture-by-numbers” application shown in [3]. In this comparison we incorporate the combined primitive (i.e. edges and the high level drawing) to allow the fine-details editing.For the manipulated image, since the original image did not contain any edges, we added and ”edge-like” layer on top of the original result. To ensure the robustness of our method to handle these hand drawn edges, we perform binary skeletonize to so that they are similar to the canny edges we’ve trained on.



J. A Step-By-Step Demonstration of Editing the Primitive

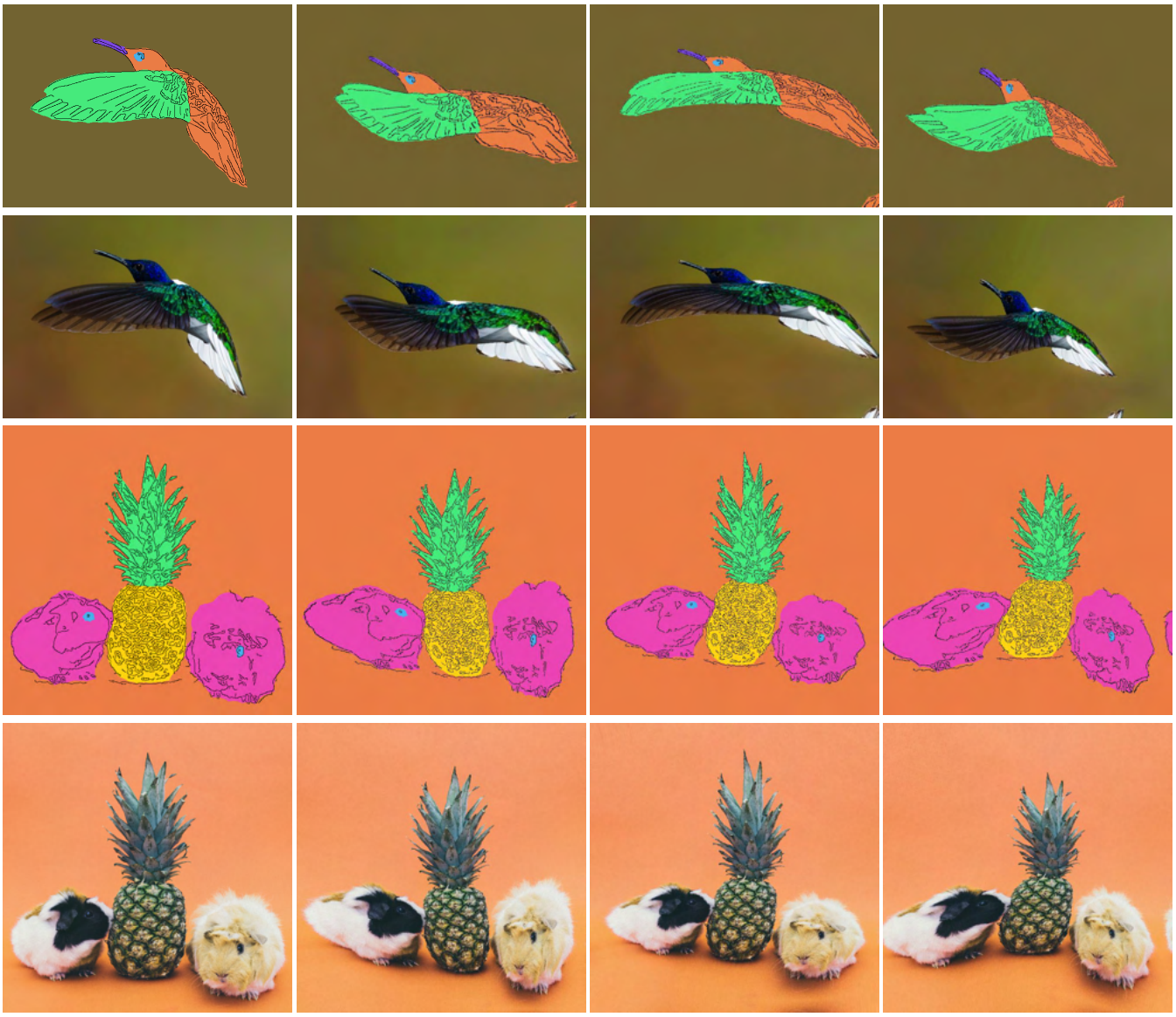
Performing complex manipulations by our method is quite easy. In this figure we present a step-by-step example of editing a primitive representation using ”Paint”. It simply requires sampling the required color and painting over the primitive image. One may also ”borrow” edges from other areas of the image to fill in empty spaces.

Step 1: Original Image	Step 2: Paint Heart
	
Step 3: Copy Edges of Trees	Step 4: Rearrange Edges of Trees
	

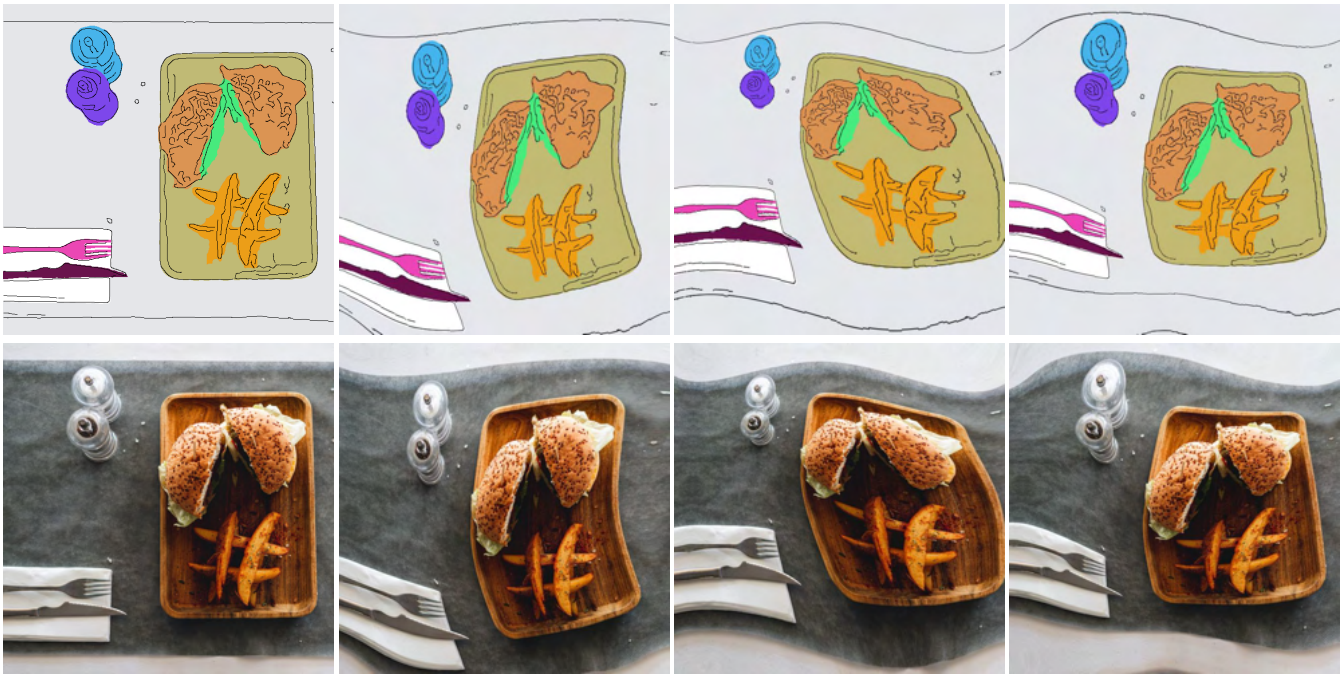
K. TPS Examples

We present several examples of original and TPS augmented images and primitives. We can see that TPS introduces complex deformations to the samples, allowing much more expressive edits than when using simple ”flip and crop” augmentations.

Original	TPS 1	TPS 2	TPS 3



Original	TPS 1	TPS 2	TPS 3



References

[1] J Canny. A computational approach to edge-detection. *Ieee transactions on pattern analysis and machine intelligence*, 1986. 9

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 9

[3] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. SIGGRAPH, 2001. 10, 19

[4] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 9

[5] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 9

[6] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 2