Unsupervised Image Segmentation With Deep Segmentation Prior

Guy Oren [1] **Shahar Azulay** [1] **Eitan-Hai Mashiah** [1]

Abstract

We investigate the use of Deep Image Prior (DIP) for unsupervised image segmentation. As in the case of supervised image segmentation, the algorithm output is an assignment of label for each pixel. However, unlike supervised segmentation, no training images or ground truth labels of pixels are available for learning. Therefore, we use DIP with low iteration volume as a segmentation prior for well known unsupervised segmentation algorithms such as K-Means (Lloyd, 1982) and also a deep learning algorithm of (Kanezaki, 2018). We show that this prior can lead to promising results for foreground-background separation as well as multi-class segmentation. code available at: https://github.com/shaharazulay/deep-segmentation-prior

1. Introduction

Image segmentation has long been a core vision problem and the focus of many research efforts. Currently, under the heavily researched supervised setting, deep convolutional networks (CNNs) are the state-of-the-art for this task. The unsupervised settings of this problem is of special interest due to the inherit difficulty in acquiring high quality labeled data representing the different object classes in a given image. The segmentation of an image can vary from a simpler settings as foreground-background segmentation to more advanced settings such as multi-class segmentation where the number of classes is not known in advance.

In light of the challenges that rise from unsupervised segmentation, we wish to mitigate them by using a "smart" (yet) simple prior. (Ulyanov et al., 2018) were first to set the terminology Deep Image Prior (DIP), when they showed that a great deal of image statistics are captured by the structure of a convolutional image generator independent of the learning process itself. The DIP network was shown as sufficient to capture the low-level statistics of a single image, without prior knowledge or supervision. Influenced by their work, we wish to set the terminology "Deep Segmentation Prior" (DSP), a variant of the DIP network with low iteration volume, that captures the attributes inside an image that are helpful for finding its segmentation.

We show that unsupervised clustering over the DSP achieves promising results for the task of foreground-background segmentation. We also show, that the DSP can serve as a powerful starting point the more complex task of multi-class image segmentation, helping in preventing over-segmentation and reducing training time.

2. Related Work

Deep Image Prior. (Ulyanov et al., 2018). In this work, the authors showed that contrary to the belief that learning is necessary for building good image priors, a convolutional image generator is sufficient to capture a great deal of the low-level statistics of a single image. To show this, they reduced reconstruction problems such as image denoising, in-painting and super-resolution to conditional generation problem, in which the only information required to solve those problems is in the degraded input image and the strcture of the network.

Double DIP. (Gandelsman et al., 2018). A follow up work to (Ulyanov et al., 2018), in which the authors used coupled DIP networks to obtain a unified framework for unsupervised layer decomposition of a single image, which they named as "Double-DIP". Their approach was demonstrated to a wide range of computer vision tasks, in particular, foreground-background image segmentation. The motivation is that it is easier for each DIP to reconstruct a specific layer of the image, in that case the foreground and background layers. A main drawback of that approach for unsupervised segmentation is the use of a heuristic saliency algorithm (Goferman et al., 2010) to produce initial guesses for background and foreground layers.

Unsupervised Image Segmentation By Backpropagation. (Kanezaki, 2018). This work offer the appealing settings of multi-class segmentation of an image without

[1]School of Computer Science, Tel Aviv University, Tel Aviv, Israel. Correspondence to: Guy Oren <guyoren1@mail.tau.ac.il>, Shahar Azulay <shaharazulay@mail.tau.ac.il>, Eitan-Hai Mashiah <eitanhaim@mail.tau.ac.il>.

Preliminary work. Under review by the Amir Globerson: gamir at post.tau.ac.i . Do not distribute.

any other training images or pixel labels. In particular, even the number of classes is not known in advance. They offer a joint learning process of image representation and pixel labels by alternately iterating between the two. To motivate a useful learning they use a superpixel algorithm, SLIC (Achanta et al., 2010), which they use to determine which pixels should have the same label.

3. Deep Segmentation Prior

3.1. Motivation

The underlying structure of a natural image has been of major recent interest as a gate for unsupervised vision tasks. Small image patch recurrence has been shown by (Glasner et al., 2009) to be a key property of a natural image that can be exploited for solving a large variety of computer vision tasks. This unique internal patch distribution inside an image was also the ground assumption used by "Double-DIP" for decomposing an image into two "simpler" layers each containing high patch similarity.

The DIP work by (Ulyanov et al., 2018) demonstrated that deep generative CNNs capture the underlying structure of a natural image, as they are trained to generate the original image from a random noise input. The DIP network is trained for a high number of iterations (the number of iterations was chosen per input image, but was usually around a few thousands) until the deep image prior recovers most of the signal while still not recovering the noise or missing patches in the image, since they are less "natural".

3.2. Prior Description

We propose a variant of the DIP optimization process which has a better fit for the image segmentation task, which we refer to as Deep Segmentation Prior (DSP). Instead of training the DIP network to recover most of the original image, we halt the optimization process at a very early stage, usually between 50-100 training iterations. Figure 1 demonstrates the learning process of the DIP network in its early stages.

This early break in the optimization process creates a DIP network that learn to reconstruct the "essence" of the input image, smoothing over fine details and complex patches in the image. This "essence" capture the nature of different surfaces in the image with high inter-patch similarity. As demonstrated in right most column in Figure 1. Therefore, the reconstructed image can be used as a good prior for the task of segmentation. As we will show in the following sections.

3.3. DSP spectral analysis

The DSP of an image is a smoother, less "detailed" version of the input image. This suggests that the DSP simply "filters out" high spectral frequencies inside the input image.

Figure 1. The learning process of DSP (shown at 10, 20, 30, 40 and 50 epochs). It can be seen that even after very few training iterations, the DSP captures a lot of the main details and surfaces in the input image.

While DSP does indeed has a higher effect over high spectral frequencies inside the input image, we show here that it is very different in nature than a standard Low-Pass-Filter (LPF) which is performed over the FFT of the input image.

Figure 2 demonstrates the spectral histogram of an image using the DSP and the LPF methods, along side the reconstructed images generated by each. It can be seen that while the DSP acts as an attenuator of high spectral frequencies, it is much more efficient in preserving many different spectral frequencies inside the image that help it better reconstruct edges which separate different objects inside the input image.

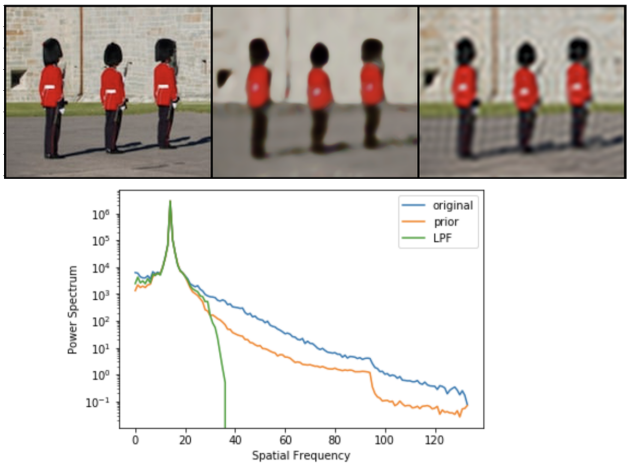

Figure 2. The DSP attenuates high spectral frequencies in the input image, but preserves the edges separating the objects in the image much better than LPF. **At the top, from left to right:** Example of input image, DSP and the image after a LPF transformation. **At the bottom:** A spectral histogram (based on the FFT transformation) of the input image, DSP and after applying LPF.

This key property, makes DSP a potentially much better fit to the image segmentation task, than a simple LPF.

4. Foreground-Background Segmentation

The DSP has characteristics that set it up to be a better starting point for the image segmentation task. It's ability to reduce high spectral frequencies in the input image and remove fine details make it a good prior for segmentation.

One of the simplest image segmentation settings is referred to as a foreground-background segmentation, which can be viewed as decomposing an image I into a foreground layer y_1 and background layer y_2, combined by a binary mask m such that $I = m \cdot y_1 + (1 - m) \cdot y_2$. This task is presented in detail in (Gandelsman et al., 2018) where the authors attempt to solve the task using an architecture they refer to as "Double-DIP" comprised of three different DIP networks, each learning to reconstruct the mask, the foreground layer and the background layer separately.

In the work by (Gandelsman et al., 2018) the input image is assumed to be constructed from two layers, each containing higher inter-patch similarity than the original image as a whole. Under this assumption each DIP network is expected to learn a separate layer, representing a "simpler" solution than any mixture between the foreground and background layers. However, due to the complex nature of converging three different DIP networks into a good image segmentation the authors apply an initial hint to the learned mask using an image saliency suggested by (Goferman et al., 2010).

4.1. Clustering With DSP

Following the characteristics of DSP, we propose a much simpler approach to the task of foreground-background image segmentation. As we saw in section 3 the DSP of an image tends to smooth the fine details inside the image while maintaining good separation between objects. This makes it a natural candidate for unsupervised clustering algorithms. We show that clustering over the DSP is a competitive match to a much more complex algorithm such as "Double-DIP", proving quality foreground-background segmentation results with a simple single DIP architecture, and with a fraction of the computational complexity.

Unsupervised image segmentation using K-Means clustering algorithm (Lloyd, 1982) was shown to be effective in some settings, for example in the work by (Yadav et al., 2015). Therefore, a question to be asked is whether the DSP is needed to segment the image, or whether K-Means clustering over the input image itself is enough.

As shown in the work by (Yadav et al., 2015), unsupervised clustering of "simple" and natural input images can achieve

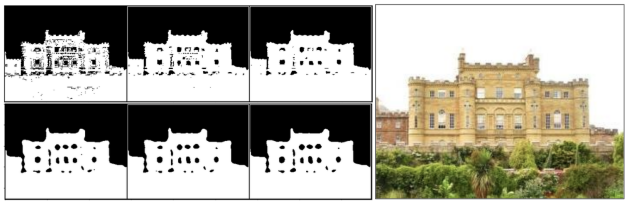

Figure 3. Deep Segmentation Prior clustering robustness to pixel window size. **On the top left**: the clustering results of the original image under window sizes of 1x1, 3x3, 5x5. **On the bottom left**: the clustering results of the DSP representation of the image under window sizes of 1x1, 3x3, 5x5. **On the right**: the original input image.

satisfying results. However, as the input image gets more detailed and diverse, clustering of the input image becomes sensitive to the selection of the number of clusters found inside the image, as well as to the size of the pixel window

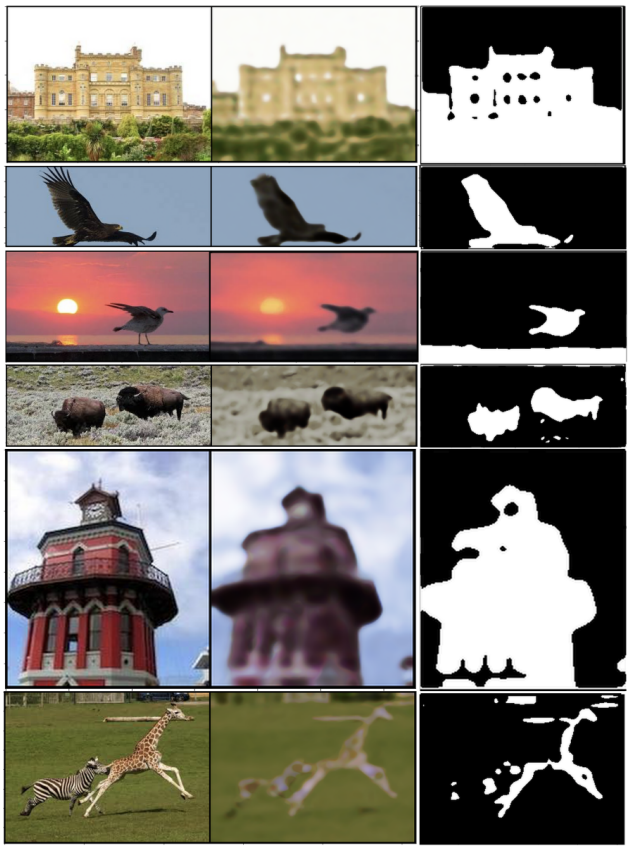

Figure 4. Clustering examples with DSP. **From left to right columns**: Input image, DSP and the foreground-background segmentation mask received after clustering with DSP. The clustering process is performed using K-Means (k=2) and a sliding window around each pixel of size 5x5 pixels.

representing each pixel in the original image.

Figure 3 shows the robustness of the DSP to the size of the chosen pixel window (representing each pixel in the clustering process) compared to the image in its original form. While the clustering of the original image tends to be too ”noisy” when the window size is too small, the results under the DSP representation are almost unchanged. This demonstrates that even for relatively ”simple” input images, the clustering of the original image is harder to manage, and is much less robust than its DSP equivalent.

Moreover, this property of DSP for clustering enable us to reduce the required window size and determine a ”good for all” window size, rather than specific window size per image. It allow us to decrease the size of the window down until a satisfying size for all images without significant degradation of segmentation quality for images that a higher window size is also suitable for them. Additionally, it also effects the computational complexity of K-Means which is proportional to the selected window size. Figure 4 demonstrates the power of the DSP to segment an image into a foreground and background layers using K-Means clustering. The results were generated by applying the K-Means algorithm (using k=2) over a DSP trained for 50 iterations to reconstruct the input image itself, where each pixel is represented by a window size of 5x5 pixels around it. We found this window size to be robust enough for all tested images.

4.2. Comparison with Double DIP

The architecture suggested by the Double-DIP algorithm is complex, relaying on a computational heavy pre-processing of the image using a computational heavy saliency algorithm by (Goferman et al., 2010) to provide an initial hint for foreground-background mask, and followed by the simultaneous training of three DIP networks for an average of few thousands training iterations.

Figure 5 show that the results achieved by the clustering of the DSP provide highly similar foreground-background segmentation to those published by the Double-DIP authors.

We also noticed that the result of the Double-DIP algorithm is highly sensitive to the initial saliency hint, which can be far from ideal under the setting of non-trivial image scene. This turns out to hurt the learning process of the DIP networks, which results in poor segmentation. Refer to Figure 6 for an illustration.

The use of the DSP representation was therefore able to closely match the original results provided by the Double-DIP algorithm and also exceed it in other complex scenarios, while using a much faster run-time and much simpler architecture, suggesting that the DSP representation can be used to simplify and harden the robustness of existing approaches for unsupervised image segmentation.

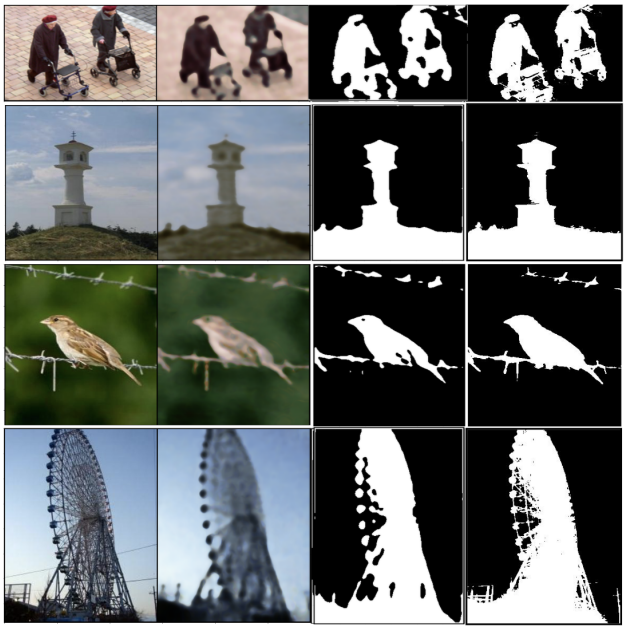

Figure 5. Foreground-background Image Segmentation. **From left to right columns:** Input image, DSP, Segmentation achieved by clustering with DSP, Segmentation achieved by the Double-DIP algorithm. All input images and Double-DIP results where taken from the original Double-DIP paper.

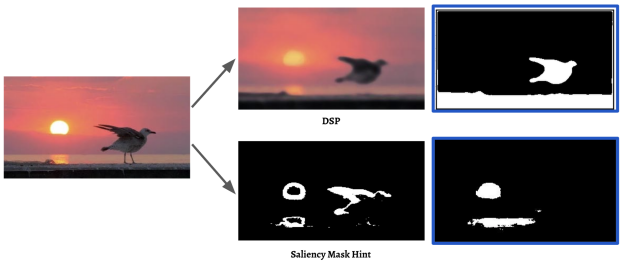

Figure 6. Foreground-background Image Segmentation Example. **On the left:** the original input image. **On the top route:** the DSP representation of the image and the resulting segmentation mask. **On the bottom route:** the saliency hint and the resulting segmentation mask of the Double-DIP algorithm.

5. Multi-class Image Segmentation

A more challenging setting for unsupervised segmentation is to segment an image into arbitrary number ($\geq$ 2) of plausible regions, which is not known in advance. The above characteristics of the DSP motivated us to check to what extent the DSP is useful for this setting. In this section we follow the work of Unsupervised Segmentation By Backpropagation by (Kanezaki, 2018) (From now on referred as USBB).

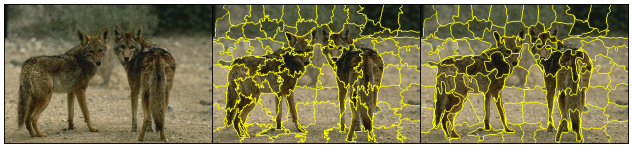

Figure 7. SLIC boundaries comparison, with up to 1000 segments and compactness of 10. **From left to right:** Input image, SLIC over input image, SLIC over DSP image

5.1. Unsupervised Segmentation By Backpropagation with DSP

The first step of the USBB algorithm is to use the superpixel algorithm SLIC (Achanta et al., 2010) to motivate the segmentation for neighboring pixels to be the same within a set of guided boundaries. This superpixel algorithm is a simple region clustering algorithm which uses the pixel values as well as the pixel distances as features. This algorithm has few hyper parameters, among them the bound of resulting superpixels and the compactness value of the superpixels. The compactness value is a trade-off between the clustering features, pixel values and pixel distances, where lower compactness give more weight to pixel values.

The properties of the DSP along with definition of SLIC, lead us to hypothesize that the SLIC algorithm over the DSP is better for the later steps in the USBB segmentation algorithm. To validate this hypothesis, we produced SLIC boundaries for both the input image and the DSP (with the same hyper-parameters). Figure 7 shows an example. We can see two main differences between the SLIC outputs - the SLIC boundaries for the input image are very noisy and there are more misalignments over objects surfaces compared to those for the DSP.

Later, we applied the rest of the algorithm steps with SLIC results over the DSP and we found out that the above two key differences lead to faster convergence of the algorithm and a more robust segmentation.

5.2. Algorithm Improvements

As part of our experiments, we noticed that the original algorithm also suffers from bad initialization and tends to get ”stuck” on bad local minimas, which often lead to under or over segmentation of the input image. We found out that this issue is not related to the use of the SLIC over DSP. Therefore, in order to address this issue, we applied two optimization techniques.

The first, we use a linear warm up phase of 100 iterations as suggested in (Popel & Bojar, 2018). This allow the algorithm to have better initialization, where it helps to stabilize the segmentation within superpixel as opposed to a very noisy initialization. See Figure 8 for an illustration.

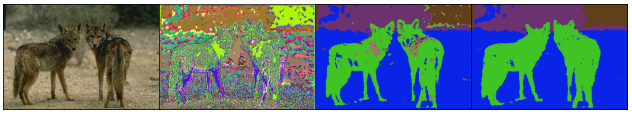

Figure 8. Warm up phase with DSP - motivation. **From left to right:** Input image, algorithm output after initialization, algorithm output after warm up phase and final algorithm output. We can see that the segmentation map after the warm up phase is a major improvement over the first initialization of the network, which is very noisy.

The second, we adjust the learning optimization schedule, after the warm up phase, by using cosine annealing scheduling as in (Loshchilov & Hutter, 2016). This scheduling allow the algorithm to ”skip” bad local minimas and allow the segmentation to be more robust.

These improvements along with the SLIC over the DSP results to an algorithm with more natural region outputs than the original algorithm. Figure 9 shows a collection of results comparing the suggested improved algorithm to the original one. More examples at the Appendix.

It is important to mention that these changes still do not solve all algorithm issues such as inconsistent results between runs or cases where there is high external patch similarity between objects in the scene. Refer to Appendix 7.2 for failure examples. A suggested way to address this is to use a dedicated loss that enforce better separation between learned regions such as exclusion loss (Zhang et al., 2018) for multiple regions. This is left for future work.

6. Conclusions and Future Work

In this work, we have shown that DSP serves as a good prior for the unsupervised image segmentation task where the external patch similarity between different regions is low. The appealing properties of the DSP for this setting enabled us to use a simple clustering algorithm such as K-Means to achieve strong results for foreground-background segmentation. DSP also proved to be effective for multi-class segmentation when combined with a relatively simple unsupervised learning algorithm by (Kanezaki, 2018). Yet, DSP alone does not suffice to mitigate all the challenges of unsupervised multi-class segmentation and requires further research.

Figure 9. Multi-class segmentation comparison. **From left to right columns**: Input image, USBB segmentation and modified USBB (with DSP, warm up and cosine annealing) segmentation.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. Slic superpixels. Technical report, 2010.

Gandelsman, Y., Shocher, A., and Irani, M. "double-dip": Unsupervised image decomposition via coupled deep-image-priors. *CoRR*, abs/1812.00467, 2018.

Glasner, D., Bagon, S., and Irani, M. Super-resolution from a single image. 2009. URL http://www.wisdom.weizmann.ac.il/~vision/SingleImageSR.html.

Goferman, S., Zelnik-Manor, L., and Tal, A. Context-aware saliency detection. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2376–2383, 2010.

Kanezaki, A. Unsupervised image segmentation by backpropagation. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1543–1547, 2018.

Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Popel, M. and Bojar, O. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70, 2018.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. Deep image prior. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.

Yadav, H., Bansal, P., and KumarSunkaria, R. Color dependent k-means clustering for color image segmentation of colored medical images. September 2015. doi: 10.1109/ngct.2015.7375241. URL https://doi.org/10.1109/ngct.2015.7375241.

Zhang, X., Ng, R., and Chen, Q. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4786–4794, 2018.

7. Appendix

7.1. Unsupervised Segmentation By Backpropagation Improvements - Comparison

Figure 10 shows a collection of examples that motivate us to use all improvements suggested above. The giraffe and shark examples show that DSP alone can transfer the segmentation from over segmentation to under segmentation (in the extreme, even to single class!). While adding cosine annealing scheduling was able to mitigate that issue for the shark it fails for the giraffe. On the other hand, adding warm up alone still suffers from over segmentation as presented in the shark (and giraffe) example. We can see that for all examples the combination of DSP with all suggested improvements lead to satisfying results.

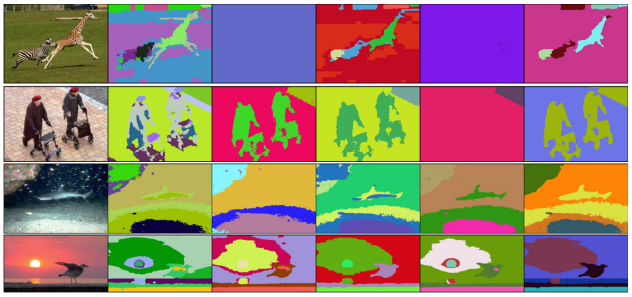

Figure 10. Improvements comparison. **From left to right**: Input image, USBB segmentation, USBB with DSP, USBB with DSP and warm up, USBB with DSP and cosine annealing, and USBB with DSP, warm up and cosine annealing segmentation.

7.2. Unsupervised Segmentation By Backpropagation Improvements - Failure Cases

Figure 11 shows some failure examples. Most of the failures originate from the fact that the external patch similarity between the objects in the scene is high. For example, the bear and eagle has high similarity with parts of the background. Other, like the seagull example expose a weak point of the DSP where the under segmentation is a result of poor reconstruction from the DSP phase. This suggests that some images require more reconstruction iterations. Another issue, which is exposed by the bird on wire example, is when the inter-patch similarity inside an object (in this case the background) still not discard all over segmented regions.

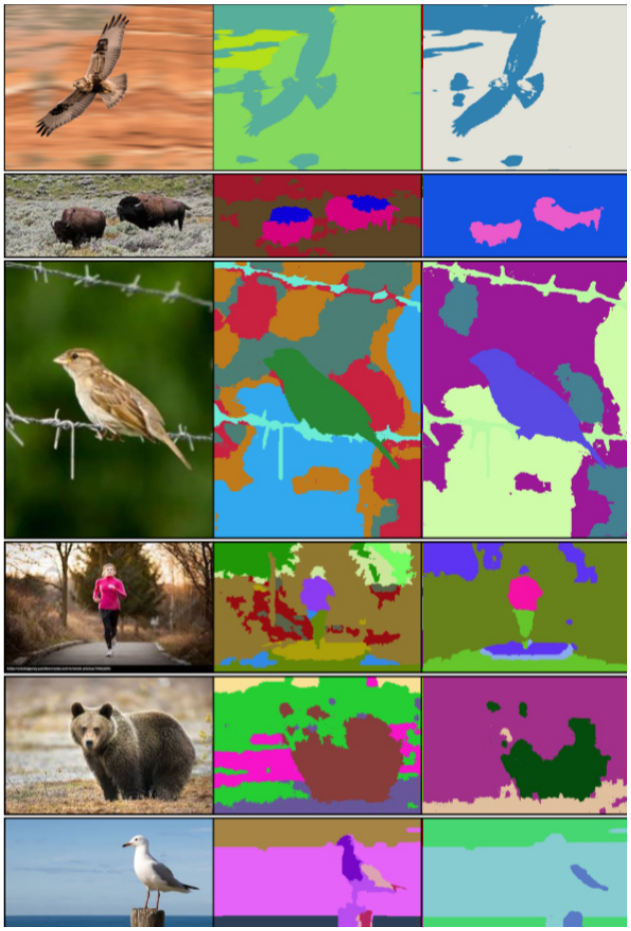

Figure 11. Failure cases. **From left to right columns**: Input image, USBB segmentation and modified USBB (with DSP, warmup and cosine annealing) segmentation.

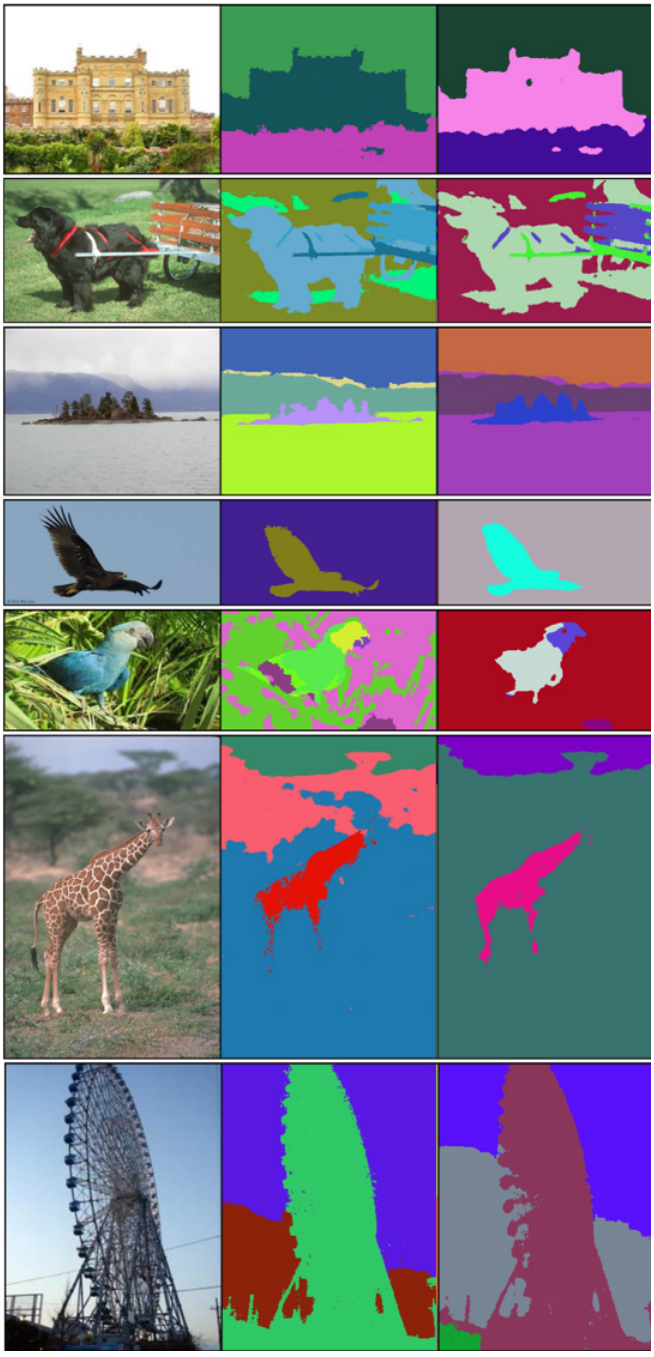

Figure 12. More examples for modified USBB. **From left to right columns**: Input image, USBB segmentation and modified USBB (with DSP, warm up and cosine annealing) segmentation.