Reproducibility Challenge @ NeurIPS 2019: Unsupervised Object Segmentation by Redrawing

Adrian M. Chmielewski-Anders[†], **Mats Steinweg**[†], **Bas T. Straathof** [†]
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden
{amca3, matsstei, btstr}@kth.se

Abstract

Object segmentation has long been a topic of great interest in the field of computer vision. One of its greatest challenges is to overcome the need for costly, large, labeled data sets with pixel-level annotations. Chen et al. [3], propose an adversarial model called ReDO, which is capable of unsupervised foreground-background segmentation in small images and can possibly be extended to multiple-class segmentation. The purpose of this report is to critically examine the reproducibility of the work by Chen et al. [3], within the framework of the *NeurIPS 2019 Reproducibility Challenge* [19]. The experiments described in this report partly corroborate the results of the original study. Moreover, the report offers a *TensorFlow 2.0* implementation of ReDO and points out several flaws in the model description in the preprint of Chen et al. [3].

1 Introduction

Semantic image segmentation is an advancing research topic. For numerous tasks, contemporary deep learning approaches are capable of segmenting high-resolution images portraying complex and varied natural scenes into well-defined, non-overlapping regions capturing the main objects in the image [2, 12, 33]. Nevertheless, these approaches require large amounts of labeled data, often with annotations at the pixel level. There is a strong urge to surmount this requirement, since the vast majority of real-world data is unlabelled and labelling can be time-consuming and costly.

The present work attempts to reproduce the contribution of Chen et al. [3], who propose an unsupervised approach to semantic image segmentation using an adversarial learning scheme. Chen et al. [3] postulate that replacing one object in an image by another should still yield a realistic image, relying on the assumption that the textures and colors of distinct objects in an image are independent of each other. The core component of the model they propose, called ReDO (ReDrawing of Objects), is a generator of new images based on regions of the input image that are segmented in an unsupervised manner. In an adversarial process [10], the redrawn regions are subsequently combined with the original image and passed to a discriminator model trained to distinguish between real and fake images, which forces the generator to produce images that are aligned to the original images. Chen et al. [3] demonstrate that this unsupervised method implicitly learns to segment objects.

As part of the replication track of the *NeurIPS 2019 Reproducibility Challenge* [19], this report presents an attempt to exactly replicate the results displayed by Chen et al. [3]. Strictly following the paper did not yield the same results as those presented in the original publication. Direct communication with the authors was necessary to expose several inaccuracies in the description of the implementation details, which will be meticulously described in section 3.4. This report makes

[†]Equal contribution

Preprint. Under review.

a three-fold contribution. First of all, it corroborates the results of Chen et al. [3]. Secondly, it offers a publicly available, working implementation in *TensorFlow 2.0*[2] in addition to the existing *PyTorch* implementation by Chen et al. [3] [3]. Thirdly, it rectifies some minor flaws in the textual and mathematical descriptions of the model in the original publication.

2 Related Work

Semantic image segmentation is a wide research field, hence, many approaches precede the work of Chen et al. [3]. To overcome the problem of requiring a large labeled data set, some approaches are weakly supervised. Common weakly-supervised approaches are to use image-level annotations instead of pixel-level annotations [7, 34], co-segmentation (i.e. identifying similar data patterns in different images) [26, 13], and the use of depth maps in addition to RGB images [25, 28]. Chen et al. [3] claim that their model is completely unsupervised, whereas the first strategy still needs labels, the second is not class agnostic, and the third cannot be used on RGB images for which no depth maps are available.

The approach of Chen et al. [3] is related to several other research areas. For example, it builds on the idea of scene composition [1, 8, 11, 32], uses tools and ideas described by Xia and Kulis [31], and is connected to approaches that also use an adversarial learning process to implicitly match a generative model to a specific task, albeit tasks that are not object segmentation [4, 20, 9, 5]. Please refer to Chen et al. [3] for a more in-depth review of these studies.

3 Reproducibility

In this section the data used to attempt to corroborate the findings of Chen et al. [3] are described, and the method applied to arrive at the results of Chen et al. are elucidated. Furthermore, all identified inaccuracies in the original paper are pointed out and justified.

3.1 Data sets

Chen et al. [3] use four different image data sets, in which all images are resized and cropped to 128×128 pixels. Three of them are publicly available and the fourth is a toy data set created by the authors themselves.

Flowers Consist of 8,819 images of flowers split into a training set of 6,149 images, a validation set of 1,020 images and a test set of 1,020 images. The images are accompanied by masks obtained via an automated method [22, 23], which are solely used as ground truths for evaluation purposes.

Labeled Faces in the Wild (LFW) Contains 13,233 images of faces [16, 14], of which 2,927 have been manually segmented and annotated [15, 17], and are used as ground truth masks. Of the masked images, 1,327 are used for validation and 1,600 for testing. The remaining unlabeled 10,296 images are used as a training set.

CUB-200-2011 (CUB) Comprises of 11,788 images of birds, of which 10,000 are used as training set, 788 as validation set, and 1,000 as test set [29].

Colored-2-MNIST This data set was contrived by Chen et al. [3] and used as a toy data set. The results on this data set were not as thoroughly described as those on the other data sets, hence, this data set is excluded from the reproduction experiments presented in this report.

3.2 Method

Chen et al.'s unsupervised generative approach to semantic image segmentation, relies on an assumption of independence between the colors and textures of the different objects in an image [3].

[2] See: `https://github.com/bt-s/dd2412-project`
[3] See: `https://github.com/mickaelChen/ReDO`

Considering images with $k = n$ classes (i.e. n-1 objects and a background referred to as n), the generative process consists of three steps: (1) *composition* – defining k different masks $\mathbf{M}^k$ (i.e. regions) in $\mathbf{I}$ based on a mask prior $p(\mathbf{M})$; (2) *redrawing* – independently generating pixels for each mask M^k based on a distribution $p(\mathbf{V}^k|\mathbf{M}^k, k)$ with a generator function of the form $\mathbf{G}_F(\mathbf{I}, \mathbf{z_1}, \ldots, \mathbf{z_n})$, where F is the object segmentation function and $\mathbf{z_n}$ a set of vectors $\{\mathbf{z_1}, \ldots, \mathbf{z_n}\}$ that are each independently sampled using a prior $p(\mathbf{z})$ for each object k; (3) *assembling* – aggregate the redrawn regions into a final image. In step 1, each pixel in $\mathbf{I}$ is assigned to a mask $\mathbf{M}^k$ corresponding to object k that has the form $\mathbf{M}^k \in \{0, 1\}^{W \times H}$. V^k in step 2 represents the pixel values of each object k and is defined by $\mathbf{V}^k \in \mathbb{R}^{W \times H \times C}$. In step 3, the images are assembled into a complete image as follows: $\mathbf{I} \leftarrow \sum_{k=1}^{n} \mathbf{M}^k \odot \mathbf{V}^k$.

To segment objects, the generative protocol described above is iteratively applied to randomly selected images in the data set. To ensure that the distribution of the output images is similar to the distribution of the training images, the object segmentation function F is learned in an adversarial fashion using the Generative Adversarial Network (GAN) scheme in which a discriminator function $D : \mathbb{R}^{W \times H \times C} \rightarrow \mathbb{R}$ learns to distinguish between fake and real images.

To encourage the model to extract meaningful masks, two constraints are applied that force it to generate non-empty masks related to the input images $\mathbf{I}$. First, to ensure that the model does not ignore $\mathbf{I}$, the assumption that the different objects are independently generated is leveraged by regenerating a single region per iteration instead of replacing all regions. This forces the generator to use original pixel values from the regions in the image that were not chosen to be reassembled. This changes the generative process to the following three steps:

$$
\begin{aligned}
&(1)\ \textit{composition:}\ \mathbf{M}^1, \ldots, \mathbf{M}^n \leftarrow F\left(\mathbf{I}\right) \\
&(2)\ \textit{redrawing:}\ \mathbf{V}^k \leftarrow \mathbf{I} \text{ for } k \in \{1, \ldots, n\} \setminus \{i\} \\
&\qquad\qquad\quad \mathbf{V}^i \leftarrow \mathbf{G}_i(\mathbf{M}^i, \mathbf{z_i}) \\
&(3)\ \textit{assembling:}\ \mathbf{G}_F(\mathbf{I}, \mathbf{z_i}, i) = \sum_{k=1}^{n} \mathbf{M}^k \odot \mathbf{V}^k
\end{aligned}
$$

The second constraint is a function δ_k that prevents the model from generating empty or constant masks. It conserves information from the latent vector $\mathbf{z_i}$ from which a region $\mathbf{G}_F(\mathbf{I}, \mathbf{z_i}, i)$ was redrawn and has the objective to infer the value of z_k given any $\mathbf{I}$. It is learned simultaneously with the generator, in a way similar to the concept of mutual information maximization in InfoGAN [4].

To extract semantic regions, the following learning objectives are used:

$$
\max_{\mathbf{G}_F, \delta} \mathcal{L}_{\mathbf{G}} = \mathbb{E}_{\mathbf{I} \sim p(data), \mathbf{i} \sim U(n), \mathbf{z_i} \sim p(\mathbf{z})} \left[D\left(\mathbf{G}_F(\mathbf{I}, \mathbf{z_i}, \mathbf{i})\right) - \lambda_z \left\| \delta_{\mathbf{i}}(\mathbf{G}_F(\mathbf{I}, \mathbf{z_i}, \mathbf{i})) - \mathbf{z_i} \right\|_2^2 \right]
$$

$$
\max_{D} \mathcal{L}_{\mathcal{D}} = \mathbb{E}_{\mathbf{I} \sim p(data)} \left[\min(0, -1 + D(\mathbf{I})) \right] +
$$

$$
\mathbb{E}_{\mathbf{I} \sim p(data), \mathbf{i} \sim U(n), \mathbf{z_i} \sim p(\mathbf{z})} \left[\min(0, -1 - D(\mathbf{G}_F(\mathbf{I}, \mathbf{z_i}, \mathbf{i}))) \right]
$$

The multiplicative constant λ_z controls the contribution of the information conservation constraint. The model is trained using the standard GAN training schema [10, 33, 21]. For the exact learning algorithm, please refer to Algorithm 1 in Chen et al. [3].

3.3 Implementation Details

The GAN setup consists of four main components: the segmentation network F, the loss conservation network δ_k, the generator G_k and the discriminator D. G_k, D and δ_k are all based on SAGAN [34], whereas F is a fully convolutional network based on [35] enriched with a Pyramid Pooling Module (PPM) [33]. The pixels resulting from F should encode local, regional and global information present in the image, which the PPM effectuates by means of several pooling layers. Conditional batch normalization is used in G_k, since it has been shown that it improves stochasticity and stimulates the encoding of style elements such as texture and colors [6, 24, 30]. With exception of F, all networks employ spectral normalization [21] for regularizing their weights, and G_k and D contain

self-attention modules [34] to capture cues from both local and non-local feature locations in the images. Furthermore, all weights are initialized orthogonally [27] and ADAM [18] is used for weight optimization. Please refer to the supplementary material in the original publication [3] for the exact architectural details of the networks and the optimal hyper-parameters.

3.4 Inaccuracies of the Original Publication

In the process of reimplementing the ReDO model from scratch to reproduce the results presented by Chen et al. [3], several descriptive inaccuracies were detected in the preprint. Communication with Chen et al. has confirmed the correctness of the inaccuracies here described.

Regarding the network architectures, Chen et al. [3] provide incorrect information about the number of feature channels in the residual blocks of D and δ. In Table 4, the number of feature channels for the four computational blocks following the Self-Attention Module have to be doubled.

In addition, the training procedure as reported in Chen et al. [3] contains some inaccuracies. In Algorithm 1, Chen et al. [3] present a faulty procedure for updating the networks. Algorithm 1 dictates that per generator update step a single region be sampled from a uniform distribution and a single $\mathbf{G}_k$ and δ_k be updated accordingly. This would alter the training dynamics, since D would be updated more frequently than the various $\mathbf{G}_k$. Therefore, all regions should be updated simultaneously instead, by iterating through all k regions per generator update step and updating each corresponding G_k and δ_k. When updating single $\mathbf{G}_k$ and δ_k, the results of Chen et al. [3] cannot be reproduced. This is most likely due to this configuration requiring a different set of hyper-parameters to achieve optimal performance. Similarly, Algorithm 1 (line 6) specifies the information conservation loss to be $\|\delta_i(\mathbf{I}_{gen}) - \mathbf{z_i}\|_2$. To be able to reproduce the results in Chen et al. [3], however, these losses must be squared. Furthermore, Chen et al. [3] implement the squared euclidean norm in this loss function as the mean of element-wise squares. Consequently, if defining the information conservation loss to be $\|\delta_i(\mathbf{I}_{gen}) - \mathbf{z_i}\|_2^2$ the recommended setting of $\lambda_z = 5$ is incorrect and has to be changed to $\lambda_z = \frac{5}{k \cdot \text{size}(\mathbf{z})}$.

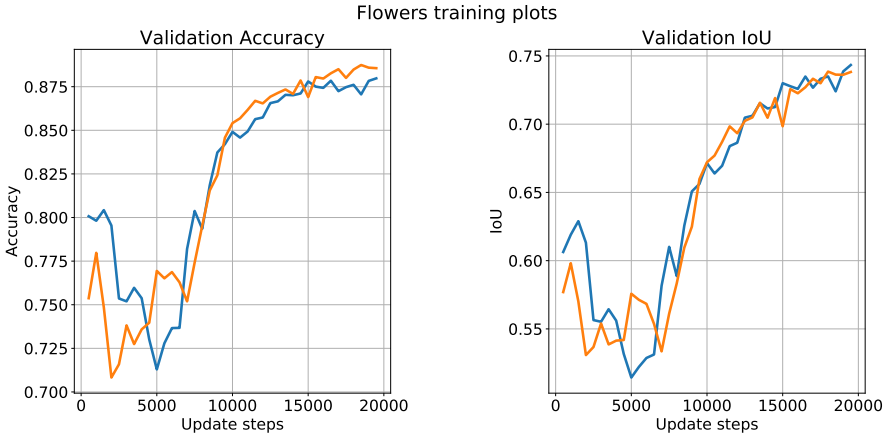

Figure 1: (left) Accuracy on the validation set for the `Flowers` data set across two training runs. (right) Intersection over union on the validation set for the `Flowers` data set. Both plots are across 19.5k update steps.

4 Experiments and Results

4.1 Experimental Setup

Chen et al. [3] train their ReDO model for the three aforementioned data sets, and report the training and testing accuracy as well as the intersection over union (IoU) between the generated masks and the ground truth masks. Besides providing examples of generated masks, Chen et al. [3] provide examples of redrawn images, though they do not report after how many update steps these images were generated. The present work reports on the same metrics and also demonstrates randomly

Figure 2: Randomly selected samples. From left to right: (1) a sampled image, (2) the true mask, (3) the predicted mask, (4-6) redrawn foregrounds for different noise vectors with the background superimposed, and (7-9) are like (4-6) but with the background redrawn instead. The noise vectors are preserved for columns across rows. Notice how the color and texture remains the same across the rows.

selected generated masks and redrawn images. Like Chen et al. [3], all experiments are carried out using an NVidia Tesla P100 GPU. Notwithstanding, using the `TensorFlow 2.0` implementation, the specified batch size of 25 had to be reduced to 18 for the `Flowers` and `CUB` data sets, and to 12 for the `LFW` data set. Aside from this exception, all hyper-parameters specified by Chen et al. [3] are adhered to. Including development trials, training results in roughly 10,000 SEK of computing costs on Google Cloud Platform.

Table 1: Original and reproduced accuracy and IoU on the test set of the `Flowers` data set, obtained after 19.5k update steps. Note: as of yet, the results for `LFW` and `CUB` have not been reproduced, hence, they are not communicated in this report. Moreover, Chen et al. [3] report their results over five runs, whereas the reproduced results are across two training instances.

Data set	Accuracy	IoU
`Flowers` (original)	0.879 ± 0.008	0.764 ± 0.012
`Flowers` (reproduced)	0.879 ± 0.003	0.736 ± 0.010

4.2 Reproduction Results

The accuracy and IoU for the `Flowers` data test set over two training instances is displayed in Table 1, where the IoU is defined as the IoU over the foreground masks. Figure 1 shows the accuracy and IoU on the validation set vs. the number of update steps and Figure 2 exemplifies generated masks, as well as random redrawn foreground and background samples.

Several training sessions on the `LFW` data set either result in the segmentation network collapsing or the network seemingly converging to accuracy and IoU scores well below two standard deviations of the means communicated by Chen et al. [3] (see Figure 3; their accuracy and IoU are 0.917 ± 0.002

and 0.781 ± 0.005, respectively). Figure 3 displays the accuracy and IoU plots vs. the training steps for various training sessions. Training on the CUB data set five times results in collapses of the segmentation network in all instances.

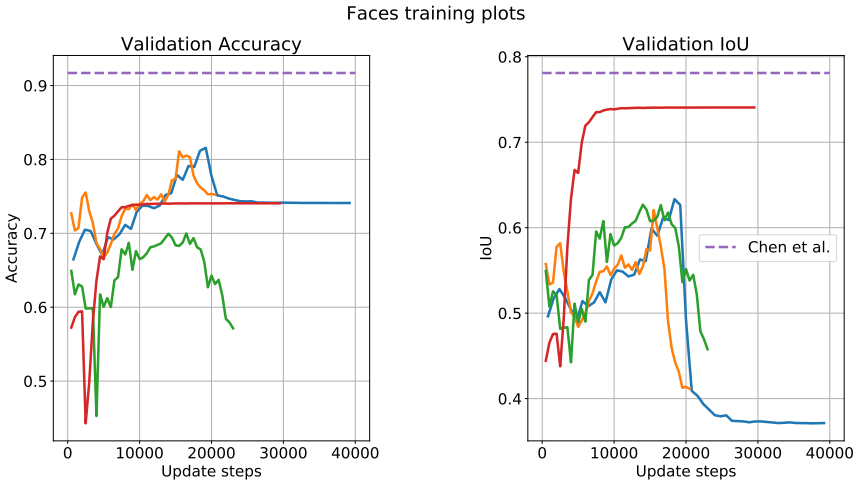

Figure 3: (left) Accuracy on the validation set for the LFW data set across 4 training runs. (right) Intersection over union on the validation set for the LFW data set. The plots are across various numbers of update steps. This is because when the masks collapsed the training was stopped. The dashed lines are the mean accuracy and IoU that Chen et al. [3] obtained on the test sets (0.917 and 0.781, respectively).

5 Analysis and Discussion of Findings

The reproduced accuracy and IoU scores on the Flowers data set displayed in Table 1 are similar to their original counterparts. This work therefore reproduces the results in Chen et al. [3] for this data set. Although only the results across two successful training sessions are displayed in Table 1, it can be concluded from preliminary results that the Flowers data set yields the most consistent and robust results, as the segmentation network rarely collapses (roughly 30% or fewer of runs). Qualitatively, the redrawings and masks look visually similar to those obtained in Chen et al. [3] with the exception of the backgrounds which are usually dark and lack detail. Although, some generated backgrounds do look more green and leafy. Training longer might produce better looking – that is to say, more binary – masks, and more detailed redrawn backgrounds.

This report does not account for more than two successful training runs on the Flowers data set.. This is because attention is also devoted to training on the CUB and LFW data sets, which take longer and are therefore more costly, and the resources for this project are limited.

The LFW data set is more complex and requires more update steps and a model with almost twice as many trainable parameters. As a result, the largest batch size that could be fit on the GPU used for the implementation described in this report was 12. The resources and time (training for 40k update steps takes approximately 48 hours) allocated for this project were only sufficient for five training sessions, none of which are close to the high performance communicated by Chen et al. [3]. The previously mentioned small batch size is the main suspect for the results not being reproducible, however, this needs to be verified. Not necessarily related to the reproducibility on this data set, but a curious observation nonetheless is that the network can still collapse after 20k iterations. Chen et al. [3] report that such behaviour is usually only observed in the earlier stages of training.

There are various potential reasons that can explain the fact that the results of Chen et al. [3] on the CUB data set could not be reproduced. First of all, communication with Chen et al. yield the finding that this data set is far more sensitive to initial conditions and has higher variance in accuracy over many training runs. Resources and time only permitted 5 full training sessions. All sessions collapsed, which seems improbable regarding the results of Chen et al. [3]. Another reason could be that since

the foreground objects of the actual images (i.e. the birds) are relatively tiny, the networks, or network hyper-parameters, may require slight alterations. However, a small hyper-parameter search of distinct values for λ_z, namely $3, 3.5, \ldots, 7$, did not yield any satisfactory results. Though further search may yield better results. As is the case with the other two data sets, the batch size had to be reduced significantly in the `Tensorflow 2.0` implementation, to be able to store all the model parameters in memory.

It is hard to pinpoint why the the results on the `LFW` and `CUB` data sets could not be reproduced, due to a plethora of potential causes. Nevertheless, since the results obtained on the `Flowers` data set with a smaller batch size as the one reported by Chen et al. [3] (i.e. 18 instead of 25) is reasonably close to their results, there is no obvious suggestion that the present `Tensorflow 2.0` implementation is flawed. Future work could attempt to establish a `Tensorflow 2.0` implementation that can use the right batch sizes. Moreover, more extensive empirical testing could conclude whether the current implementation is capable of matching the results communicated by Chen et al. [3].

6 Conclusion

In spite of difficulties encountered in reproducing ReDO, this report generally corroborates the findings by Chen et al. [3], by means of both quantitative metrics and qualitative redrawings on the `Flowers` data set. Presently, a conclusive statement as to whether the work of Chen et al. [3] is reproducible is forestalled, since more extensive training and testing is required on the `CUB` and `LFW` data sets. In addition to testing the reproducibility, this report points out several reporting errors in Chen et al. [3], some of which have been attended to in the latest version of their paper. The inaccuracies in the original reporting caused issue, as did vagueness in referenced works and somewhat surprising lack of publicly available, *correctly* written reference implementations of popular network components such as spectral normalization, conditional batch normalization and instance normalization. Without accessibility to the ReDO code, and communication with Chen et al., the feasibility of reproducing the results in Chen et al. [3] would be significantly more difficult given limited resources.

References

[1] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[3] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. *arXiv preprint arXiv:1905.13539*, 2019.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[5] Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pages 4414–4423, 2017.

[6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[7] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 642–651, 2017.

[8] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pages 3225–3233, 2016.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, 2015.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] Klaus Greff, Raphaël Lopez Kaufmann, Rishab Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *arXiv preprint arXiv:1903.00450*, 2019.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[13] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Co-attention cnns for unsupervised object co-segmentation. In *IJCAI*, pages 748–756, 2018.

[14] Gary B Huang and Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, pages 14–003, 2014.

[15] Gary B Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[16] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[17] Andrew Kae, Kihyuk Sohn, Honglak Lee, and Erik Learned-Miller. Augmenting crfs with boltzmann machine shape priors for image labeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2019–2026, 2013.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Rosemary Nan Ke Hugo Larochelle Joelle Pineau Koustuv Sinha, Jessica Forde. Reproducibility challenge @ neurips 2019, 2019. URL `https://reproducibility-challenge.github.io/neurips2019/`.

[20] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017.

[21] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[22] Maria-Elena Nilsback and Andrew Zisserman. Delving into the whorl of flower segmentation. In *BMVC*, volume 2007, pages 1–10, 2007.

[23] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[24] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[25] Trung T Pham, Thanh-Toan Do, Niko Sünderhauf, and Ian Reid. Scenecut: Joint geometric and object segmentation for indoor scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[26] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 993–1000. IEEE, 2006.

[27] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[28] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[29] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[30] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016.

[31] Xide Xia and Brian Kulis. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*, 2017.

[32] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. *arXiv preprint arXiv:1703.01560*, 2017.

[33] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[34] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3800, 2018.

[35] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.