Context Associated Object Removal from Images

Ansh Jain
jain98@wisc.edu

Kaushal Rai
kaushal.rai@wisc.edu

Avinash Kumar
kumar243@wisc.edu

Kriti Goyal
kriti.goyal@wisc.edu

Abstract

Object removal is an open research problem with many applications and algorithms producing unsatisfactory results. The outputs are unpredictable, of poor quality, and semantically incoherent. In our work, we tackle one of these issues and try to create semantically coherent outputs by removing the context associated with the object as well. We focus on shadows and aim to develop a framework to automatically remove objects with their corresponding shadow. Traditionally the task of object removal was done using semantic segmentation coupled with an inpainting technique. However, such a framework fails to capture the context associated with the removed objects and therefore has residuals such as reflections or shadows still remaining in the image. Furthermore, free-form-based inpainting models perform well in removing objects from an image but require human annotators to provide the masked input for such models.

In this project, we develop an end-to-end framework to tackle these problems. We perform an in-depth analysis to conclude that a framework consisting of LISA coupled with DeepFillv2 shows the best result on the said task. Further, we introduce a heuristic function to improve the quality of the output and obtain excellent results.

1. Introduction

Object removal has immense application in the field of video and photo editing. Several tools and software online have been developed to handle these tasks. The task of object removal has traditionally been handled using two approaches. The most common method to handle this task is by generating semantic segmentation of the input image, masking the desired segment, and then using an inpainting algorithm to remove the desired object. A significant drawback of such methods is that they fail to capture the associated context for the removed object, as seen in Fig 1.

Since object removal is closely related to image inpainting, a lot of recent research has focused on handling this task using an inpainting algorithm. Traditional inpainting algorithms only allowed masked bounding boxes for inpainting. However, modern frameworks have given us the ability to do free-form inpainting, which is more suitable for the task of object removal. These methods, however, require a lot of manual intervention. A human annotator has to manually mask the region of the images they want to remove and then feed the altered image as an input to the inpainting framework. This method causes a lot of overhead and hence is unsuitable.

(a) Original image

(b) Car removed

Figure 1: Object removal task leaving behind shadow

In this project, we try to develop a framework to handle the task of object-context removal with minimal human intervention. For the scope of this project, we focus only on'Shadows' as the context of an object. To handle the task of context-associated object removal, we first use the Light Guided Instance Shadow-object Association (LISA) [7] framework to map the objects with their respective shadow. The model is trained on a new dataset called Shadow-OBject Association(SOBA) which consists of fine-grained annotations of objects and their shadows, as shown in Fig 2. The crucial aspects of the architecture are further

explained in section 3.

Figure 2: Instance shadow masking in SOBA dataset

We then augment the pipeline with a free-form image inpainting framework. We experimented with two recent free-form image inpainting algorithms for our project. Firstly, we use the DeepFillv2 framework [10], which uses a gated convolution network. We compare the results of this framework with another free-form inpainting framework, deep image prior [5]. Deep image prior suggests that the structure of a generator network is adequate to capture a large number of low-level image statistics. It uses a randomly-initialized neural network as a constructed prior in typical inverse problems like denoising, superresolution, and inpainting, with excellent results.

With the above two modules, we found that due to the inaccuracy of the segmentation module, we obtained weird artifacts in the final output after inpainting. Therefore, simply coupling these two segments was not enough for object-context removal as LISA failed to produce very fine-grained segmentation maps and could not classify pixels near the edge of the objects accurately. This can be seen in Fig 3 where removing even a simple pole does not provide realistic output. We implement our own heuristics between the two modules and present the result to further mitigate these problems. We experiment with breadth-first search algorithm and Superpixel generation algorithms to enhance the quality of masked images formed as the output of the LISA framework. The results show that adding our heuristic functions dramatically improves the quality of the image formed.

To summarize, our contributions to the project are threefold:-

- Firstly, we create an end-to-end framework that automatically removes an object and its corresponding shadow with minimal human intervention.
- To improve the quality of the results, we propose and implement different heuristics that improve the segmentation result from LISA. With simple algorithms, we can see immense improvements in the output.
- Finally, in this work, two different inpainting algorithms are compared, which approach the problem differently. We compare the qualitative results from both and present our analysis for the same.

(a) Original image

(b) Output Image

Figure 3: Result of simply coupling LISA and DeepFillv2

2. Related work

Instance Shadow Detection Traditional semantic segmentation do not take context of an object into account [11], [4]. However, earlier works in Computer Vision models that aimed at finding or removing shadows from other objects used edge color separations and physical illumination. Wang et al. [7] aimed to find individual shadows through the image along with the associated object that cast them. They prepared a new dataset SOBA - Shadow

OBject Association with shadow instance masks and shadow-object association masks. The authors introduced LISA - Light-guided Instance Shadow-object Association which systematically finds (i) individual shadow and object instances, (ii) shadow-object associations, and (iii) light direction for each association. It uses ConvNet to extract semantic features from the image. They formulate SOAP - Shadow Object Association Precision as a quantitative evaluation metric and demonstrate the applicability of their work in shadow removal editing and detecting the direction of the light. We use LISA as the backbone for our architecture.

Inpainting Image inpainting is an essential vision task for modification, restoration, quality enhancement, etc. Although it has been around for several years, the recent developments in image processing techniques and the need for digital image editing has given automatic inpainting surge in popularity. We surveyed several methods [8], [3], [5], [10] and used two of them to compare outputs.

Ulyanov et al.'s Deep Image prior [5] serves as a bridge between the two inpainting methods - learning-based which uses CNNs, and non-learning based using self-similarity and other handcrafted image priors. The authors show that it isn't just the learned image prior from a large dataset that helps deep convolution networks get good results, but infact, it is the structure of the model that can capture low-level image statistics prior to any pre-training. For common inversion problems such as inpainting, they show randomly-initialized neural network also gives good results when used as a handcrafted image prior. They test on two inpainting settings, one, randomly drop 50% of the pixels, and two, mask large holes. In the second task, which is relevant to our work, the authors admit since this is a non-learning based method, Deep Image Prior isn't expected to work on highly semantic hole inpainting such as faces, but they showed some examples where it outputs approximately desired result. They concluded that having a deeper architecture network and including skip-connections that improve segmentation tasks such as recognition are unfavorable for inpainting using Deep Image Prior.

Feedforward networks with deep convolution have been the go-to for intricate inpainting holes involving faces, complex scenes, and objects. These networks learn from large datasets to retain semantics and transfer content to new images. However, because of convolutions filters' inability to distinguish invalid from valid pixels in the image, they often render outputs with visual artifacts with ghosting effects, blurry reconstruction, and color discrepancy. Yu et al. [10] proposed gated convolution for free-form image inpainting. Their model learns the feature selection method dynamically across all layers for each channel and spacial location. Gated convolution works better when masks are arbitrarily shaped and can have complex conditional inputs such as sparse sketch beside the RBG channels. They also introduce SN-PatchGAN - a patch-based discriminator. These help improve color consistency and leads to overall higher-quality outputs than previous state-of-the-art on benchmark datasets such as CelebA-HQ faces and Places2 natural scenes. Gated convolution work builds up from "Generative Image Inpainting with Contextual Attention" [9] by the same authors. In this work, they experimented with a two-stage network. In the first stage, to rough out the masked content, a dilated convolution network is trained with a reconstruction loss. In the second, contextual attention is used to generate unknown patches from known. This approach showed good and promising outputs but was very slow.

Superpixel Grouping pixels into grids with meaningful, perceptually distinct boundaries forms the basis of several image manipulation tasks. Multiple algorithms have been proposed for this purpose such as [6], [2], etc. Achanta et al. [1] compared five state-of-the-art superpixel methods based on adherence to the boundary (arguably the most important property), processing speed, impact on segmentation performance, and quantitative. They also introduced an adapted k-means clustering approach - Simple linear iterative clustering (SLIC). The algorithm besides the image just needs one parameter, 'k' the expected approximate number of superpixels, each about the same size. Empirically, they concluded SLIC outperformed other surveyed methods. In this work, we use SLIC along with our heuristic algorithm to produce masks.

3. Method

The pipeline developed in this work, as shown in Fig. 4 can be divided into three major components. These components and their architectures are explained in subsequent subsections.

3.1. Instance Shadow Detection (ISD)

The first module utilized is instance shadow detection (ISD) [7]. The model developed by the authors in [7], called Light-guided Instance Shadow-object Association, can automatically predict shadows, objects, their associations, and the direction of the light source for an input image. It produces pixel-level segmentation and the bounding boxes for each segmented entity. It learns from jointly minimizing the loss of all the outputs. For our purpose, we filter out and utilize only the essential outputs of the model, i.e., the pixel level segmentation of the object, its shadow, and their association. For each object-shadow association, we save a sep-

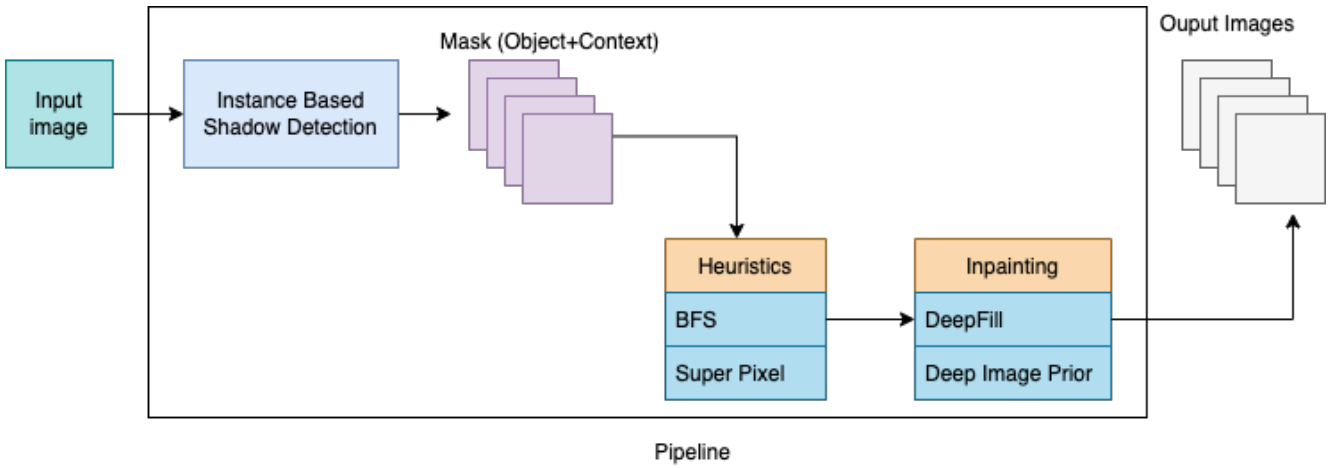


Figure 4: Proposed Pipeline

arate mask, by masking out the single object and its shadow, and keeping the remaining image as background. This enables the user to select which object needs to be removed, and the corresponding mask can be fed into the inpainting component.

The outputs produced from the pre-trained model suffer from the drawback that they fail to classify pixels accurately on the boundary of the object/shadow as shown in Fig 5. This results in nonsensical outputs from the pipeline as certain artifacts remain in the image and mask used as input to the inpainting module.

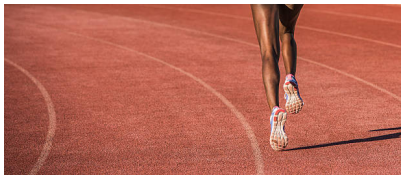
Figure 5: LISA segmentation drawback (the mask is not able to capture the entire shadow or even the entire object

3.2. Heuristic for improvement

To mitigate the issue from LISA, we implement post processing algorithms on the output of ISD and compare their effects.

- Breadth-First Search (BFS) - Initially to test our hypothesis that the inpainting module would perform better if the entire object is masked properly i.e. with no remaining edges, we implement a simple BFS and mask the neighboring pixels up to a particular distance from the actual object/shadow mask. We vary the distance using the size of the object masked. For example, if the original object mask has 'n' pixels, we run the BFS up to a distance of x% of n. x is varied between 0.067, 0.1, 0.2 and we found the best result to be with x = 0.2. As expected the pipeline worked significantly better and was able to mask the entire object when given sufficient margin Fig. 6. However, this algorithm does not take into account the semantics of an object and just increases the size of the mask. This could have the side effects of covering some neighboring objects that should not be removed and also would be unable to capture different shapes in the remaining region.

- Superpixel - To incorporate semantics into the algorithm and improve the output quality we replace the BFS algorithm with superpixel generation. We first cluster neighboring pixels into a superpixel using the method Simple Linear Iterative Clustering (SLIC) [1]. This method adapts k-means clustering to create superpixels that are related semantically in low-level features. Then, we check each superpixel and count the number of pixels masked above a certain threshold we mask that entire superpixel. This algorithm can generate a minimal mask that can cover the entire image as shown in Fig. 6. The heuristic only masks semantically related pixels and avoids some of the drawbacks of the previous approach. However, for our project, we were unable to find a generic parameter structure that worked well for each image, and therefore for further experiments, we use the BFS algorithm itself.



(a) Original Image

(b) Mask with BFS margin = 0.1%

(c) Mask with BFS margin = 0.33%

(d) Mask with Superpixel k=160

Figure 6: Optimal masking algorithm comparison

3.3. Inpainting

The third module is image inpainting, which takes in the input image and the corresponding mask and fills the masked region with information relating to the surrounding context. We also study different inpainting algorithms and choose the following two architectures that employ vastly different concepts to perform the task. The common aspects of both the architectures are that they are capable of handling free-form input masks and claim to perform well only when a maximum of 20-30% of the input image is masked.

- Deep Image Prior - The paper relies on the "structure" of the network to capture low-level information before any learning. The authors fit a generator network to a single image that is degraded (for example masked object, noisy image, etc.). This method learns separately for each test image and therefore is extremely slow for testing each image, however it saves the cost of training on a large dataset. The problem is formulated as an energy minimization problem.

$$\theta^* = \arg\min_{\theta} E(f_\theta(z); x_0), x^* = f_{\theta^*}(z)$$

For the inpainting task, the energy term is

$$E(x; x_0) = ||(x - x_0) \odot m||$$

where $\odot$ is the element wise product, x_0 is the initial image, x^* is the final output image, and x is iterative image being generated by the model. z is the initial random noise and $f_\theta(z)$ is the function that maps noise to the image space. Also, $m \in \{0,1\}^{H \times W}$. We utilize the same network which the authors have utilized for the "library" image, as devising a separate network for each test image is beyond the scope of our project.

- DeepFillv2 - We use DeepFillv2 for comparison with deep image prior. The main aspect of the network is the gated convolution, which enables information from the unmasked pixels to slowly propagate to the masked pixels. Also, they use a contextual attention branch to improve the quality of the output. Finally, the generated output along with its mask is sent to the discriminator which categorizes each patch as "real" or "fake".

4. Experiments and Results

We have evaluated the proposed pipeline by varying different critical components of the architecture. Our pipeline contains two key modules dealing with segmentation and inpainting tasks. Hence, we need to evaluate these components individually and examine the pipeline's final results. The upcoming subsections provide a detailed overview of the experimental setup based on segmentation, inpainting, and heuristics techniques.

4.1. Datasets and Pretrained Models

We have conducted experiments on two different pretrained models for our segmentation task. The baseline model used for comparison is referred from the following github[1] repository. LISA uses the SOBA data set for training which was created by the authors themselves. Similarly, for the inpainting task also, we have used two different pre-trained architectures, namely DeepFillv2 and Deep Image Prior. The latter does not require any training dataset, whereas DeepFillv2 uses Places2, and CelebA-HQ faces as datasets.

Since our proposed pipeline comprises two distinct tasks, different sets of evaluation metrics are required to examine the results. Table 1 summarizes the evaluation metrics used for different models and tasks based on our proposed idea. For ISD, the results are evaluated using Shadow-Object Average Precision (SOAP), which is based on the idea of traditional average precision (AP) with the intersection over

[1] https://github.com/sujaykhandekar/Automated-objects-removal-inpainter

Table 1: Evaluations Metrics

Task	Model	Metrics
Segmentation	Base Model	PSNR
	ISD	SOAP
Inpainting	DeepPrior	PSNR
	DeepFillv2	l_1/l_2

union (IoU). However, the metric considers the shadow-object association while determining valid positive samples. Similarly, for quantitative analysis of DeepFillv2 and DeepPrior, we have used l_1/l_2 and PSNR, respectively. In terms of model specification, the pre-trained DeepFillv2 uses 4.1M parameters along with an inference time of 0.21 seconds per image on single NVIDIA(R) Tesla(R) V100 GPU and 1.9 seconds on Intel(R) Xeon(R) CPU @ 2.00GHz for images of resolution 512 × 512 on average, regardless of mask size.

4.2. Evaluation Scenarios

Based on the previous discussion over possible configurations, we have conducted the experiments for four different combinations. Table 2 Summaries the pipeline for three different cases. We have applied different heuristic methods with different hyperparameter values((margin and partitions) in each case.

Table 2: Experiment Cases

	Pipeline
Case 1	Base Model
Case 2	ISD + Heuristic + DeepPrior
Case 3	ISD + Heuristic + DeepFillv2

4.2.1 Result Case 1: Base Model

We have used an architecture built upon deeplabv3 and edge-connect for our baseline model. This architecture also uses the similar idea of image segmentation and inpainting to achieve the object removal task. However, it does not consider the context associated with the removed object. As shown in Fig 7, the baseline could not remove the associated shadow, but our proposed pipeline was able to remove the object along with the context. The results produced by this baseline mainly fails due to the poor performance of object and context association in the segmentation task.

4.2.2 Results Case 2: ISD and DeepPrior

We applied different heuristic methods to finetune the results from ISD and make them more suitable for the inpainting task and then apply the Deep Image Prior Architecture. We have observed that the results delivered from this pipeline are not that aesthetic. Even though the model successfully associated the object and its shadow in most cases, the inpainting results were less realistic than the DeepFillv2 (Fig. 8).

(a) Original Image

(b) Baseline

(c) Our Model (ISD+DeepFillv2+BFS)

Figure 7: Qualitative comparison with the baseline

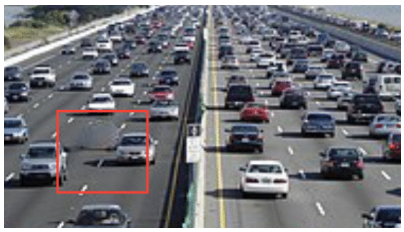
4.2.3 Results Case 3: ISD and DeepFillv2

We evaluated this pipeline by incorporating the heuristic functions described in the previous section to compare various scenarios. Based on the results, we have a few observations.

Firstly, the architecture delivers well irrespective of the number of primary objects in the image. We tested with a single object and its shadow and multiple objects and their

(a) Original Image

(b) Result using deep prior



(c) Result result using DeepFillv2

Figure 8: Analysis between result of Case 2 and Case 3

shadows; in both cases, our model was able to remove the objects and their associated shadows (Fig. 9).

Secondly, the results showcase a relationship between the size of the object we are trying to remove and the output quality. It was observed that the larger object size leads to poor quality output. However, output quality is quite realistic if the background is less complex (solid or less texture). The primary reason for this observation was that the pre-trained DeepFillv2 model delivered good results while completing only 30% of the image (Fig 10).

Finally, results also show that in most cases, the heuristic approach based on superpixel performs much better if the object shape is complex than the BFS-based approach. However, superpixel algorithm is difficult to generalize across images due to many hyperparameters. Fig 6 shows that with superpixel, we can achieve minimal mask to remove an object with context.

5. Conclusion

Removing objects entirely from a scene along with associated context is an open problem. The effect an object has on its environment is complex and may include shadows, reflection, and other indentations. For example, when a person jumps from a trampoline, the cupping on the surface of

(a) Original Image

(b) Single Object

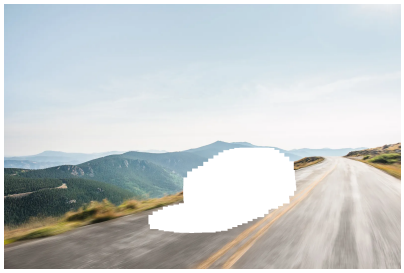
(c) Multiple Objects

Figure 9: Removing objects along with associated shadow

the trampoline must be removed too along with the person to make the altered scene believable. In this work, we explored shadow-object removal. Shadows by themselves are complex phenomena to handle - in cases such as multiple shadows, crooked shadows due to obstructions, overlapping shadows with surrounding objects, soft shadows, etc. We hope to pragmatically improve the cases mentioned above.

Our pipeline consists of 3 modules - one, instance shadow detection, two, heuristic, and three an inpainting module, DeepFillv2, or Deep Image Prior. The advantage of the pluggable nature is that we could easily swap our individual components in case new research is published that is better suited. The drawback however is that since it is not trained in one go, the model doesn't itself learn for example which segmentation gives the best output. Qualitatively, we determine pipeline consisting of Instance Shadow Detection + BFS + DeepFillv2 produces the most desirable outputs.

(a) Original Image



(b) Masked Image

(c) Final Output

Figure 10: Removing objects along with associated shadow

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 3, 4

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 3

[3] H. Liu, B. Jiang, Y. Xiao, and C. Yang. Coherent semantic attention for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4170–4179, 2019. 3

[4] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo. Swin transformer V2: scaling up capacity and resolution. *CoRR*, abs/2111.09883, 2021. 2

[5] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. 2, 3

[6] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *European conference on computer vision*, pages 705–718. Springer, 2008. 3

[7] T. Wang, X. Hu, Q. Wang, P.-A. Heng, and C.-W. Fu. Instance shadow detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1880–1889, 2020. 1, 2, 3

[8] L. Wu, C. Zhang, J. Liu, J. Han, J. Liu, E. Ding, and X. Bai. Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1500–1508, 2019. 3

[9] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 3

[10] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 2, 3

[11] Y. Yuan, X. Chen, X. Chen, and J. Wang. Segmentation transformer: Object-contextual representations for semantic segmentation. 2019. 2