Better Image De-occlusion Using Similar Images

Guoyuan An [*1] **Heeju Wi** [*1] **Junyong Park** [*1] **Saelyne Yang** [*1]

Abstract

Understanding natural scene image is an important task in computer vision. However, image de-occlusion remains as challenging which aims to recover and complete the invisible parts of occluded objects in an image. In this report, we replicate a paper which proposes a self-supervised learning for scene de-occlusion (Zhan et al., 2020). Then, we improve the quality of resulting de-occluded images of the model by exploiting similar images to a given occluded image. To do so, we first implemented an image search API that extracts the most similar image to a given image from the web. Then, we designed and implemented a novel convolution network that combines occluded image, modal mask, predicted amodal mask, predicted image resulted from the given model and a reference image retrieved by the image search API. Our approach shows that the quality of image de-occlusion improves with the similar images. We believe that it can benefit many applications such as image inference or recomposition.

Figure 1. An example of image de-occlusion. (a) Original image with the table occluded by a card wallet. (b) Image de-occlusion recovers occluded part. (**Note that this is just an example to show the concept; not actual de-occlusion)*

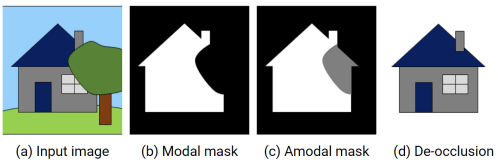

Figure 2. The general process of image de-occlusion. With an (a) Input image with the house as the target object, (b) the modal mask is first obtained using common image segmentation methods. (c) Then the amodal mask, which is a whole mask including invisible part, is inferred (d) Finally, the color for the invisible parts are filled in to complete the whole object.

1. Introduction

1.1. Background

Natural scene image understanding is an important task in computer vision. The recent advent of deep neural networks as well as large-scale annotated datasets allowed many image understanding tasks such as object detection (Girshick et al., 2013; Ren et al., 2015; Wang et al., 2019) and semantic segmentation (Girshick et al., 2013; Dai et al., 2016; Chen et al., 2019). These methods can successfully segment and create masks for objects within an image. However, they mostly focus on segmenting the visible parts of the objects. A real-world scene is composed of multiple objects, resulting in some parts of the objects being occluded by other objects. Image de-occlusion, or amodal mask and content generation is the process of solving such problems. It aims to recover and complete occluded parts of an image (Figure 1).

The general process of image de-occlusion is shown in Figure 2. With an input image and a target object (Figure 2(a)), modal masks, which is a mask of visible part of the object is obtained using common image segmentation methods (Figure 2(b)). Then, the amodal mask, which is a mask of a whole object including invisible parts is obtained (Figure 2(c)). Finally, the color for the invisible parts are filled in to complete the whole object, completing the de-occlusion task (Figure 2(d)).

1.2. Related work

We surveyed three related works in the field of scene de-occlusion. The first paper *SeGAN: Segmenting and Generating the Invisible* (Ehsani et al., 2018) used a GAN-like

[*]Equal contribution [1]School of Computing, KAIST, Daejeon, South Korea.

Final report for 2020 spring CS570, KAIST.

network consisting of a Segmentor, Generator, and a Discriminator. The Segmentor takes the occluded RGB image and the modal mask as input and outputs a predicted amodal mask. The Generator then takes that amodal mask to fill in the RGB pixels of the occluded areas and returns the final de-occluded RGB image. The Discriminator was used to make the finals images as realistic as possible. The second paper *Visualizing the Invisible: Occluded Vehicle Segmentation and Recovery* (Yan et al., 2019) also used a GAN-like network with a similar workflow. Their focus was on de-occluding images of vehicles with the Discriminator networks classifying whether the generated images are of a real vehicle.

The final paper *Self-Supervised Scene De-occlusion* (Zhan et al., 2020) is the paper that we replicated in this work. The main contribution of this paper is the self-supervised method of training scene de-occlusion. By erasing random parts of images and considering that as occlusion, they did not need any labeled de-occlusion dataset. Further details are described below.

1.3. Replicated paper

The paper *Self-Supervised Scene De-occlusion* (Zhan et al., 2020) proposed an image de-occlusion framework with self-supervised method, without ordering and amodal annotations as supervisions. We chose this as our replicated paper as it is the first attempt to use self-supervised method for image de-occlusion.

An overview of the proposed framework in the paper is as follows. With an input image and the associated modal masks as input, the framework first recovers ordering information between objects in the image to extract which object occludes or is occluded by other objects. With the ordering information, it progressively does amodal completion to fill the mask of invisible parts and content completion to fill the actual content (color) of the invisible parts under the guidance of amodal predictions.

The de-occlusion is achieved by two novel networks, PCNet-M and PCNet-C (Figure 3). First, the mask completion is achieved by PCNet-M. With an input instance A and a random instance B, PCNet-M is trained by switching two cases: Case 1 (A erased by B) follows a partial completion mechanism where PCNet-M is encouraged to partially complete the input instance A. Case 2 prevents PCNet-M from over completing A (Figure 3(a)). Next, the content completion is achieved by PCNet-C. PCNet-C erases intersection between A and B from A and learn to fill in the RGB content of the erased region. It also takes in A erased by B as an additional input (Figure 3(b)).

The proposed framework generates amodal masks that are as accurate as ground truth and resulted in seamless content completion. However, we could note that the content completion of the proposed method is still not very accurate. For example, in Figure 4 we can see that in the image of the drawer, there still exists traces of the chair. Therefore, we aimed to improve the quality of content completion of the proposed framework.

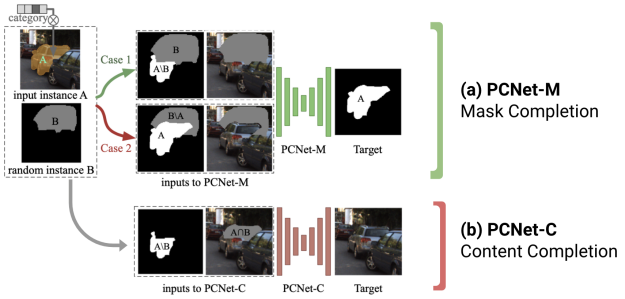

Figure 3. (a) PCNet-M for mask completion and (b) PCNet-C for content completion in *Self-Supervised Scene De-occlusion*

2. Improvement Approach

To improve the quality of de-occluded images resulted from *Self-Supervised Scene De-occlusion*, we obtain and utilize images similar to the de-occluded images. Then we use two methods (image stitching and convolutional network) to increase the performance of the replicated paper. Our contribution could be summarized as:

- We propose a **novel approach** to de-occlusion task by using **similar images**, which are proved to be effective through our experiment results.
- We propose two approaches (**convolutional neural network** and **image stitching**) that use the retrieved similar images to increase the performance of content completion.

The limitation of de-occluded images from the paper *Self-Supervised Scene De-occlusion* is that it doesn't clearly remove occluding objects when filing the content that was occluded. This is because the model does not know the original appearance of the occluded object. Thus, we thought that providing visual information of the target object would be a key for improving the quality of de-occlusion. To achieve this, we came up with the idea of **exploiting similar images to a given image**. By using similar images, we expect that we can identify and extract details of each object that may not appear in the given input image but appears in the similar images.

We built an Image search API that automatically searches for images with a given image on Google and extracted the

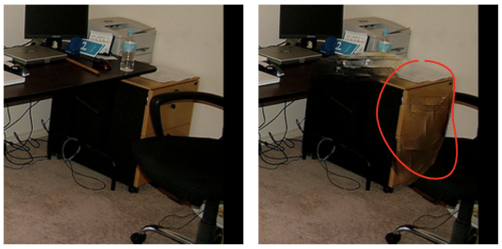

Figure 4. De-occlusion result of *Self-Supervised Scene De-occlusion*. (a) Original image where the drawer is occluded by the chair. (b) Resulted image after de-occlusion. The drawer is awkwardly filled with just similar colors.

Figure 5. Images in our new dataset. (a) The black-colored area is erased from the original object image for the evaluation. (b) You can see the erased area is filled with some awkward pattern by the original model. (c) The reference image is obtained using our image search model and the original erased image.

most similar image to the given image based on similarity algorithm. We call the most similar image to a given image as a **reference image**, which will be used as an additional input to our newly proposed model. We discuss how do we use the reference images to improve the results below.

2.1. Convolutional model approach

We designed a **convolutional neural network** to incorporate the reference images along with the outcomes from the existing model. We trained it to understand significant information from reference images when performing de-occlusion. If the reference images are images of objects similar to the objects we are trying to de-occlude, they can provide information about the missing parts of the object. We aimed to test whether a convolutional model could learn to extract the significant areas from the reference images in order to accurately filling in missing pixel values.

Similar methods of transferring information from one image to another can be seen in other fields such as image style transferring and clothing transferring. Gatys (Gatys et al., 2016) uses a CNN model to transfer the style of one image to a second image while keeping the contents of the second image the same. Raj (Raj et al., 2018) uses a convolutional GAN model to transfer clothing images onto images of a person. Likewise, we aim to see if we can transfer the visual appearance information of missing or occluded areas of an object from the reference images.

2.2. Image stitching approach

We also tried the **image stitching** technique to solve the problem. Image stitching is a process of combining multiple images with overlapping view. If we could retrieve a perfect reference image, de-occlusion can be done by filling in the corresponding patch of the reference image using image stitching. We leave details on image stitching approach in the Appendix, as it was not our primary focus.

3. Dataset

3.1. COCOA dataset

Our replication paper used COCOA dataset, which is subset of COCO dataset(Lin et al., 2014) with amodal annotations(Zhu et al., 2015). It contains images, which have various objects, and the modal and amodal masks of each objects with pairwise ordering. The training data has 2500 images with 22163 objects and the test data has 1323 images with 12753 objects.

3.2. Our new dataset

Since our reference image search API takes about 12 seconds for each image, we ran the process asynchronously and built a new dataset using the original COCOA dataset to reduce the time cost. Our new dataset contains a reference image for each object image.

Our data consists of modal mask, amodal mask, ground truth images from the original COCOA dataset, predicted amodal mask, predicted image from the pre-trained model of the replicated paper, eraser, erased mask which are artificially erased part of the given object image to test our color completion results, and reference image that we retrieved by our image search API. Figure 5 shows a sample data of our new dataset. Since there were several images that we could not get any results from our API due to a big image size, we used only 9440 images for our data out of 22163 objects in original COCOA dataset.

4. Experiments

4.1. Image search API

We built an image search API to get similar images from the web. Here, we used Google Image Search which allows search by images and returns similar images to the

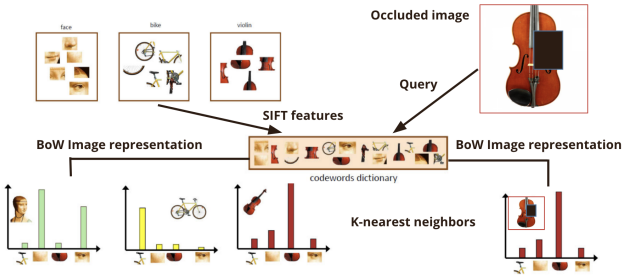

Figure 6. The detail of our re-rank process. We build the codewords dictionary using the SIFT features and prepresent every image using BoW. The most similar images of the occluded image could be found by K-nearest neighbors.

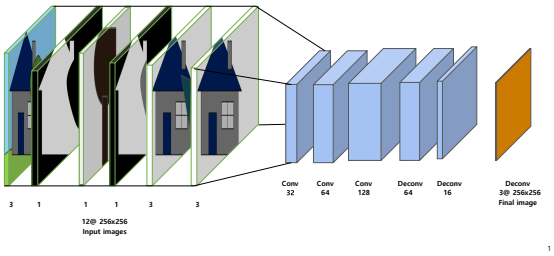

Figure 7. Our proposed convolutional model

image. Our image search API first uploads a target image to imgbb.com, a website for hosting images, and then does Google Image Search using Selenium, a web framework for crawling. Finally, the API gets images in search results.

4.2. Similar image retrieval technique

Because Google image searching has a diverse metric used to compare similarities of images, not all the results are necessary useful for our de-occlusion target. We therefore do a second search from the google results to filter out the useful reference images. Here we introduce the image retrieval technique used in this project briefly.

An image retrieval system reflects every image into the latent space using an embedding vector, so that the similarity of two images could be easily checked by calculating the Euclidean distance between their corresponding vectors. We first find SIFT key points in all images and then summarize the frequency of each key point in every image using a pooling function BoW, as shown in Figure 6. Then we use the K-nearest neighbors to find the top-k most similar images with the occluded image.

4.3. Convolutional Model

Our convolutional model consists of 3 convolutional layers and 3 deconvolutional layers. The width and height of outputs does not reduce as we want to keep the original structure of the image as closely as possible. The inputs to the model are 1) RGB image of the occluded object, 2) modal mask of the object, 3) mask of objects occluding the occluded object, 4) predicted amodal mask of the object (from PCNet-M), 5) predicted de-occluded RGB image (from PCNet-C), and 6) reference image. These 6 images are concatenated to form a 12 channel input to our model. The final deconvolutional layer of the model outputs a 3 channel final image from which we mask the occluded area and combine it with the original image to produce the final image. Details of our model are shown in Figure 7.

We trained two models with the same structure. The first model used images from the image search API as the reference images during training. The second model used modified images of the ground truth images as the reference images during training. To prevent the model from learning that the reference images are ground truth, we transformed the ground truth images in complex ways including randomly altering the gamma values, adding padding, cropping, rotating and shearing.

This third model is included after the PCNet-C model of our replication paper. The model was implemented in PyTorch. Training was done for 100 epochs which took around 6 hours with a RTX 2080Ti GPU. We used a batch size of 16, a learning rate of 0.001, and a stochastic gradient descent optimizer. For the loss, we measured the mean squared error on the pixels of the occluded area that needs to be filled.

5. Results

5.1. Replication Results

Since the original model evaluated in various applications including ordering recovery, amodal completion, we followed their steps. The original paper used two datasets, KINS(Qi et al., 2019) and COCOA, so we used these two datasets and their train, test splits.

We report the ordering recovery performance of the original model and our replicated model on COCOA and KINS in Table 1. Our replicated model performance is similar especially on COCOA. We also evaluated amodal completion on ground truth modal masks of the original model and our replicated model, as also shown in Table 1. Our replication model performance is very similar to the original model. MIoU(Mean Intersection over Union) is the most frequently used metric for segmentation and object detection. IoU(Intersection over Union) of two objects is calculated as dividing the overlapping region by the combined region, and MIoU is the average of IoU per image. These metrics

which were used in our replication paper however do not consider the colors.

Table 1. Replication of ordering recovery and amodal completion

MODEL	COCOA	KINS
ORDERING RECOVERY		
ORIGINAL MODEL	87.1	92.5
REPLICATED MODEL	87.112	88.803
AMODAL COMPLETION (%MIOU)		
ORIGINAL MODEL	81.35	94.76
REPLICATED MODEL	81.346	94.356

5.2. Our improved model results

5.2.1. SIMILAR IMAGE RETRIEVAL RESULTS

Figure 8 shows a few examples of the reference images we obtained through our API. The top row shows images containing occluded objects (with the occlusion shown in black) and the bottom row shows the results of the API. We can see that for the left two images, the resulting reference images are quite similar to the query images and contain the visual contents of the occluded areas.

For some query images however, as shown in the right two images, although the reference images were quite similar to the query images, they did not contain useful visual contents for the occluded areas. In the third image in Figure 8, the occluded object is a paper stuck on a wall. The reference image also contains papers on a wall and the whole image is quite similar to the query image, but intuitively, the contents of the reference image does not seem to be useful for filling in the occluded area.

Sometimes, the object occlusion was too large that the image search API wasn't able to get a similar image at all. The rightmost query image in Figure 8 is an image of a cup noodle with a major area of it occluded. We can see from the reference image result that the API is not able to get a cup noodle image but instead gets an image of a hamburger.

5.2.2. DE-OCCLUSION RESULTS

Quantitative evaluation

Due to the lack of evaluation on color completion using metrics in our replication paper, we choose and implement the image similarity metrics following the paper (Chaur-Chin Chen & Hsueh-Ting Chu, 2005; Dengsheng Zhang & Guojun Lu, 2003). We adopted four metrics which were widely used for image similarity measure; euclidean distance, manhattan distance, chord distance and Mean squared error.

We trained our two models to compare with the original replicated Pcnet-C model. We call the model that is trained on altered ground truth images as reference images the

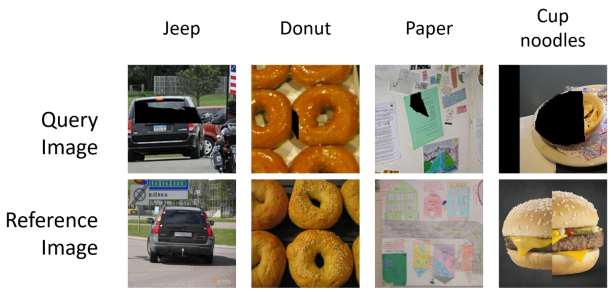

Figure 8. Examples of the reference images retrieved through our Google image search API.

Table 2. Quantitative evaluation on color completion using four image similarity metrics; euclidean distance, manhattan distance, chord distance and Mean squared error. We compared our two improved models with the original replicated model. We put the mean value of each metric of the whole test data.

MODEL	EUC.	MANH.	CHORD.	MSE.
ORIGINAL	87.936	134.257	0.481	87.937
ALTERED-GT	94.630	**118.936**	**0.371**	94.630
SIMILAR-REF	90.961	137.365	**0.357**	90.961

"Altered-GT" model, and the model that is trained on images from Google images as the reference images the "Similar-Ref" model. For the test, since we don't have ground truth data, we made our test data which contains 500 random object images from COCOA. Those images are randomly occluded once more by us, to make ground truth image, which is the image before our artificial occlusion.

Table 2 shows our evaluation results on color completion of the occluded images. The smaller metric value means the images are more similar to their ground truth. When comparing the models with Manhattan distance, "Altered-GT" model outperformed the original model. Moreover, both our improved models outperformed the original model by the Chord distance. However, you can see there's no perfect relation between the metrics and this is quite common issue when you compare the images. So as the image similarity research do, we evaluated using multiple metrics.

Our research problem was the de-occluded objects images doesn't seem good, especially on occluded part, using the model of the target paper. Our improved models showed better results with some metrics for image completion. We can say that our improved models completes de-occluded images better than the original model in some way. However, the image similarity metrics cannot reveal whole human's perception on images, so the best way is to check the results visually which is shown in the next section.

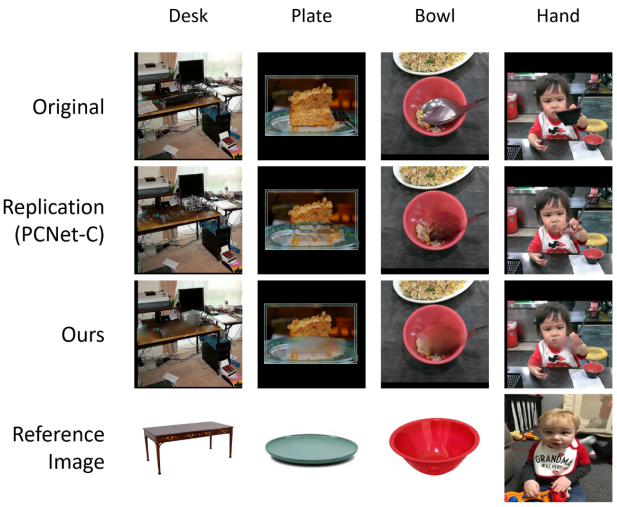

Figure 9. Few examples of the output image of our model compared to those of the replicated paper (PCNet-C). The reference images of the first two objects (Desk and Plate) were modified by us in order to get more accurate results.

Qualitative evaluation

We made a dataset of images where occlusion isn't done artificially, but is instead done by actual objects in the image scene. Figure 9 shows a few examples of our output images for the testing dataset. Because we do not have the ground truth data for de-occlusion, we compare out output images with those of our replication paper (PCNet-C). For the first three images (Desk, Plate, Bowl), we can see that our results are slightly more accurate than the results of PCNet-C. When the target object is a desk, a perfect de-occlusion should be able to erase all items on the desk and show a plain brown desk. We can see that the desk in our image is slightly more clean that the desk in the PCNet-C's image. With the plate and the bowl in the second and third images, similarly to the desk image, the results of our model show a cleaner plate and bowl, indicating better de-occlusion.

However, like the last image in Figure 9, if the reference image does not contain the visually significant contents to fill the de-occluded area, our model is not able to accurately de-occlude the image, and instead the target areas is blurred. This shows that our model is weak at filling in small details of images and also needs good reference images.

6. Discussion

We can see through the quantitative and qualitative results that our convolutional model performed image de-occlusion slightly more accurately than the original replicated PCNet-C model. However, because our model is influenced by the reference image, in cases where the reference image is inaccurate, our model performed poorly. Such examples are shown in the rightmost images of Figure 8 and Figure 9. In this section we describe some limitations of our work along with potential improvements for better de-occlusion.

6.1. Similar image retrieval

In order to get reference images that are similar to the target objects, we built a simple Google image search API and then reranked the images in terms of their similarity with our query image. However, as shown in the Results section, if the query images are significantly occluded or if target objects are part of the background of an image, the resulting reference images either had totally different objects or did not contain visually significant contents. This naturally made the reference images useless, if not harmful, when performing de-occlusion.

One possible way of improvement in order to get more accurate reference images with helpful visual contents would be knowing what the object in target is. If the type of the target object can be inferred from the occluded image, then the reference image would be an image of the same or similar object, making it more likely to have the significant contents. Also, when doing the image search, instead of using the whole occluded image, adjusting the image so that the target object is the main focus would help with cases where the the target object is less visible or is part of the background.

6.2. Convolutional model

The model we used in our work was a simple convolutional model consisting of convolutional layers and deconvolutional layers. However, the results of our model were not significantly better than the original replicated PCNet-C model. Because the results of the PCNet-C model were already fairly accurate, to make the de-occlusion even more accurate, the model would need to be able to on the details of the missing parts. We believe that our model currently is not able to completely make use of the given reference image and understand the significant contents in it, because of the lack of enough data and also the poor quality of the reference images obtained. By training the model on a larger dataset, with more accurate reference images, we think the model would be able to perform even better de-occlusion under diverse circumstances.

Another problem with our current model is that we do not have a discriminator to force the model to produce images that seem natural. Although not included in this paper, in some results, the model had filled in pixels of the occluded area too extensively. A GAN model was used for the PCnet-C model of our replication paper, showing that if we add a similar discriminator model to ours, the resulting output images could be more natural and accurate.

References

Brown, M. and Lowe, D. G. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

Chaur-Chin Chen and Hsueh-Ting Chu. Similarity measurement between images. In *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, volume 2, pp. 41–42 Vol. 1, 2005.

Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C. C., and Lin, D. Hybrid task cascade for instance segmentation, 2019.

Dai, J., He, K., Li, Y., Ren, S., and Sun, J. Instance-sensitive fully convolutional networks, 2016.

Dengsheng Zhang and Guojun Lu. Evaluation of similarity measurement for image retrieval. In *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, volume 2, pp. 928–931 Vol.2, 2003.

Ehsani, K., Mottaghi, R., and Farhadi, A. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6144–6153, 2018.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Qi, L., Jiang, L., Liu, S., Shen, X., and Jia, J. Amodal instance segmentation with kins dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Raj, A., Sangkloy, P., Chang, H., Lu, J., Ceylan, D., and Hays, J. Swapnet: Garment transfer in single view images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 666–682, 2018.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 06 2015. doi: 10.1109/TPAMI.2016.2577031.

Wang, J., Chen, K., Yang, S., Loy, C. C., and Lin, D. Region proposal by guided anchoring, 2019.

Yan, X., Wang, F., Liu, W., Yu, Y., He, S., and Pan, J. Visualizing the invisible: Occluded vehicle segmentation and recovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7618–7627, 2019.

Zhan, X., Pan, X., Dai, B., Liu, Z., Lin, D., and Loy, C. C. Self-supervised scene de-occlusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, June 2020.

Zhu, Y., Tian, Y., Metaxas, D. N., and Dollár, P. Semantic amodal segmentation. *CoRR*, abs/1509.01329, 2015. URL http://arxiv.org/abs/1509.01329.

A. Appendix - Image Stitching

Most approaches to image stitching require nearly exact overlaps between images and identical exposures to produce seamless results, but some stitching algorithms actually benefit from differently exposed images by doing high-dynamic-range imaging in regions of overlap. We tried using similar images for image stitching techniques. We describe details below.

A.1. Modified UKBenchmark dataset

The original UKBenchmark (https://archive.org/details/ukbench) has 2550 categories and each category has 4 images that show the same object. We crop all the images so that every image only show a part of the object (then every image could be seen as occluded). We divide the modified dataset into 7650 occluded images and 2550 occluded images for training and testing respectively. Our task completes the occluded testing images by searching and using the reference image from the training images.

A.2. Experiment

Given an artwork with damaged or missing parts, image stitching fills in the corresponding patch of reference image to make it complete. The most important things here is how to find the matching points of two images and how to keep spatial invariant (keep same direction) of the the content of two images.

Inspired by the paper (Brown & Lowe, 2007), we use SIFT features and **BFMatcher** in opencv to find the matching points of two images. In order to keep the spatial invariant of them, we use **RANSAC** (Fischler & Bolles, 1981) to compute Homography and then use it to warp perspective. Check the detail of image stitching technique in (Brown & Lowe, 2007).

A.3. Results

Figure 10 shows the performance of our image stitching approach. Given an occluded image, our search engine could correctly find the reference image from the training data, and then our image stitching engine successfully completes the occluded patch. Notice that our completion result is very similar with the origin image. This result shows that if there is images of same instance among the dataset, image search and stitching could replicate the occluded image well.

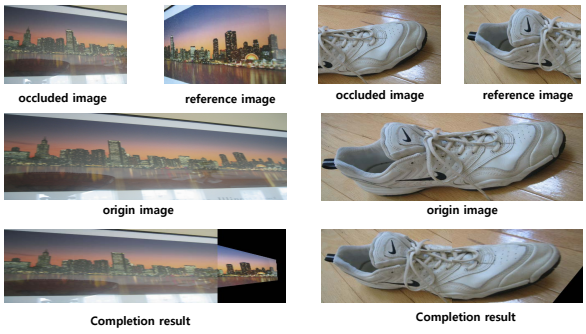

Figure 10. Two examples of the image stitching result. Occluded image is a part of the origin image. The reference image is retrieved from the training dataset. Our stitching result correctly completes the occluded part of the origin image.