Diminished Reality with Inpainting Methods

Kehan Wang, Jingyi Song
University of California, Berkeley
Berkeley, CA
{wang.kehan, jingyi_song}@berkeley.edu

Abstract

Diminished Reality(DR) has not been a deeply explored Computer Vision field, but its potential is limitless as Augmented Reality(AR). Hoping to bring more attention to DR, we combine state-of-the-art segmentation and inpainting methods and developed an online DR-ready platform. We also proposed a new inpaintor, SwinUPer, that uses Swin Transformer[10] as vision-backbone. We achieved better l1-loss and PSNR than state-of-the-art inpainting methods, trained and evaluated on Places365 dataset. Our code is publicly available at `https://github.com/Jason-Khan/dr_inpaint`.

1. Introduction

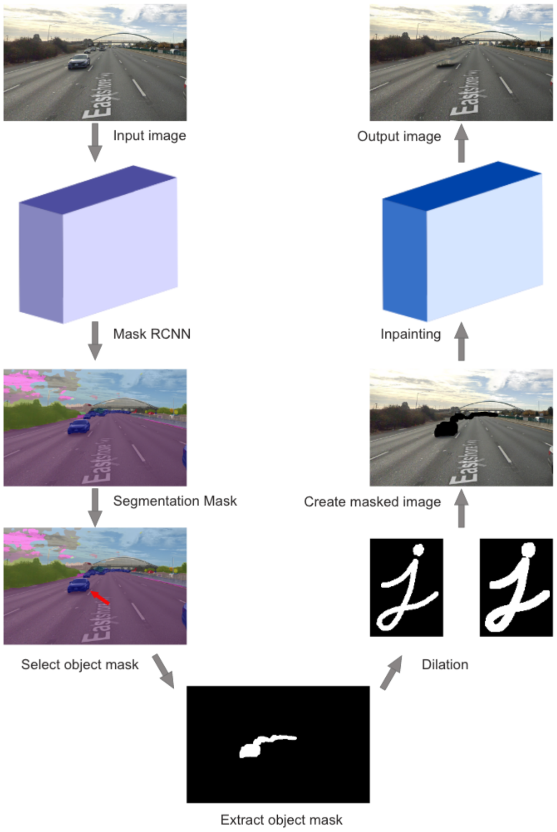
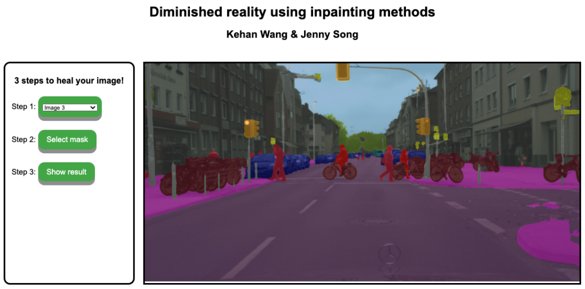


Figure 1. Diminished Reality Model

Augmented Reality(AR) is an interactive experience of the real-world where parts of the visual display are enhanced using Computer Vision methods. Existing AR technologies take advantage of the camera feed and add visual effects or virtual objects into the feed to enhance the 2-D reality. AR can greatly benefit people's everyday life by offering information about the real-world in real-time. It can be used for gaming entertainment, for navigation instructions or even medical operations. AR has been a huge hit to the mobile application industry, and as rumors of Apple AR glasses float around, more attention has been drawn to the AR industry. However, AR cannot achieve its full-potential without being able to manipulate the actual reality - Diminished Reality(DR), where people can choose to remove an object from a given scene. Being able to remove objects from a real-life scene amplifies the effect of AR, and immerse people in the real Virtual Reality. DR has many applications, such as virtual furniture replacement and wire removal in videos. There have been very few DR-ready applications available online, and of those available, the incorporated methods to achieve DR are out-of-date and could benefit an update from new state-of-the-art Computer Vision methods. We target one DR pipeline, hidden view generation, and combine state-of-the-art semantic segmentation and inpainting methods to revamp the available DR tools.

[11] summarized 5 different ways of achieving DR: background observation, scene tracking, detection of region of interest, hidden view generation, and composition. They can be further grouped into two category - a) with information of the obstructed background; b) without such information. A lot of the DR applications recover the background by considering its structures and geometry. Some compose



Diminished reality using inpainting methods

Kehan Wang & Jenny Song

3 steps to heal your image!

Step 1: Image 3

Figure 2. Online DR tool

different angles of views to piece the background together. [18], [17], [6] use inpainting to fill in the background. We focus on the inpainting approach. Inpainting is the process of repairing or completing an image. It is achieved by synthesizing the missing parts using information from the rest of the image. Others use texture transfer to complete the image, while [18], [17], [6] adopted a more modern approach of using deep learning on GANs to learn the distribution of the hole given the rest of the image.

As [15] propose Transformer, the first sequence transduction model based entirely on the attention mechanism, it has been raising increasing attention. [2] incorporated the attention mechanism in Natural Language Processing(NLP), and devised Masked Languagen Modeling(MLM) as a task of pretraining the language model. The task of MLM is highly similar to the task of image inpainting - MLM replaces 15% of the text to be predicted by the language model, and inpainting asks the vision model to predict parts of an incomplete image. With the recent development in vision transformer backbone, we ask ourselves, is it possible to train a Transformer vision model for inpainting, similar to how BERT is pretrained on MLM? We propose a new inpainting model using semantic segmentation by Mask-RCNN to create a masked image, and a general GAN pipeline that adapts Swin Transformer [10] and UperNet [16] as our generator. Because of the lack of training resources, we were only able to partially train our model on Nvidia Tesla K80 GPUs. Despite the lack of trainin, our non-optimized model still achieved 0.171 l1-loss, 22.69 PSNR and 0.807 SISM on Places365 dataset[19], outperforms the state-of-the-art inpainting method[18], which evaluated at 0.184 l1-loss, 21.79 PSNR and 0.805 SISM.

Our main contributions are:

1. Develop an online DR tool2 that uses the state-of-the-art semantic segmentation and inpainting methods.
2. Propose a transformer-based inpainting method, Swin-UPer.

2. Related Work

2.1. Diminished Reality

Diminished reality is a set of methods that visually remove, hide and see through real objects in real time. Diminished reality technology is used to implement diminish, see-through, replace and inpaint functions[11]. For the inpaint technique, the basic approach is to replace the pixels from the real object to pixels based on surrounding patches. There is no guarantee of a perfect recovery to the original background. However, recent work from [7] and [8] has generated quality results for curved and multiple surfaces.

2.2. Semantic Segmentation

Image segmentation is a computer vision task in which we label each pixel of an image with a corresponding class of what is being represented. The segmentation does not differentiate objects of same class, which differs from instance segmentation. Semantic segmentation cannot be resolved by standard convolution network followed a fully connected layer because the length of the output is variable, since the number of occurrences of the objects of interest is not fixed. A naive approach to solve the problem is to take different regions of interest from the image, then use CNN to classify the object within the region. However, because the object of interest can have different locations and multiple aspect ratios, a sliding window is needed to select huge number of regions and this could accumulates and consumes a lot of time. Therefore, algorithm like R-CNN[4], which propose selective search algorithm[14] to extract regions from the image, largely reduced the number of region needs to be looked at. Fast R-CNN[4] improves it further by feeding the image into a CNN first than identify region of proposal from the generated feature map. As region proposals become bottlenecks in the algorithm, Faster R-CNN[13] is developed which uses a network to learn the region proposals instead of search. Algorithm like YOLO[12] also achieved good performance by using a single convolution network predicts the bounding boxes and the class probabilities. Nowadays, Mask R-CNN[5] is state of art semantic segmentation technique that could generates robust results.

2.3. Inpainting Methods

[6] proposed using a global context and a local context discriminator to help the generator learn image completion methods that both consider the structure of the entire image and also the details closer to the mask. It is a one-stage inpaintor as only one generator is used between masked image and inpainted image. Instead of using the discriminator to learn the local details, [17] propose a two-stage generator. The first generator generates a coarse inpainting result, where the second-stage generator focus on the overall struc-

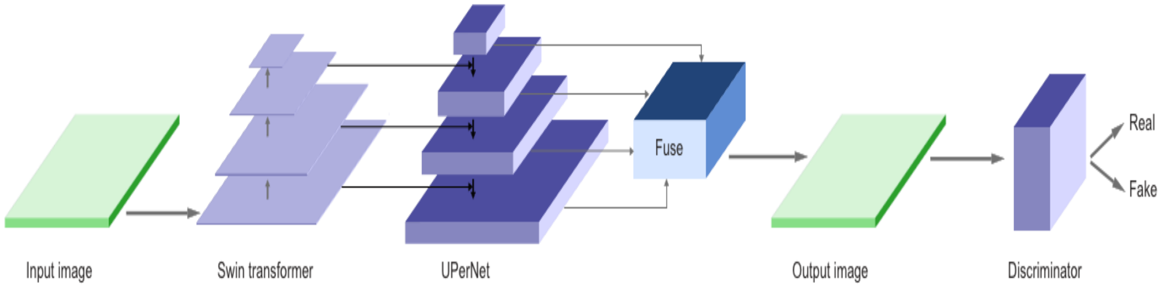


Figure 3. SwinUPer Inpaintor Pipeline

ture of the image and inpaint the high-frequency details. [18] continued the two-stage generator appraoch, and proposed the idea of gated convolution, a learned partial convolution that ignores the masked parts of the image when extracting features. It also proposes a loss function called SN-PatchGAN loss, which combines PatchGAN loss with the spectral-normalized discriminator. It also brings up the free-form inpainting challenge, where the masks provided to the inpainting method need not to be a bounding box any more - it can be of any form that pen strokes can compose.

2.4. Transformer

Transformer model architecture is proposed in [15], where repeated attention and multilayer-perceptrons serve to embed temporal data. BERT[2] is the first model to use Transformer on NLP tasks. It takes word embeddings and add a positional embedding to the sequence of the word to capture the temporal information. It also proposes to pretrain BERT on masked language modeling and sentence completion to achieve self-supervised learning. ViT[3] is the first to propose the idea of a transformer vision-backbone. It breaks the input image into patches and sees each of the input patch as a embedded word. By adding positional encoding for each patches, it replicates what BERT[2] does for words, and proposed masked patch prediction as a self-supervision pretraining method for vision-transformers. Swin Transformer[10] is a recent transformer vision-backbone that performed better than many state-of-the-art convolutional models on various vision tasks. It proposes Shifted Windows method to create a hierarchical structure that generates feature maps of different scales and encapsulate different scales of attention field.

3. Background

Attention mechanism in Transformers[15] is computed as a scaled dot-product. It utilizes similarity to measure the amount of attention required for each feature embedding. With input queries Q, keys K of dimension d_k and values V of dimension d_v, the Attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

In order to create masking effect in attention, we set our masked pixels to 0 so the dot-product with masked pixels return 0 similarity, pushing the first layer of our Transformer to ignore the masked pixels.

Feature Pyramid Network(FPN) is first proposed in [9]. It employs a bottom-up feature pyramid to extract semantic features of different scales, and use all of them in a top-down feature pyramid that makes multiple decisions as the semantic scale decreases. FPN extracts a finer feature map and is great for tasks that require classification at a smaller granularity, such as semantic segmentation and inpainting. UPerNet[16] uses the idea of FPN and incorporates different decoding heads for different tasks at a pixel level, such as object detection, material detection, and texture detection.

4. Proposed Method

4.1. Diminished Reailty

Our proposed pipeline for the DR experience is as such Fig.1:

1. Use Mask R-CNN to generate a segmentation mask for our user to select from.
2. User chooses an object to remove.
3. Extract object mask using Breadth-First Search.
4. Dilate object mask.
5. Create masked image by superimposing the object mask on the original image.
6. Use Inpainting method to complete the image.

In some cases, the semantic segmentation mask does not cover the edges of the object, which disturbs our masked image's content distribution. This creates visual artifacts

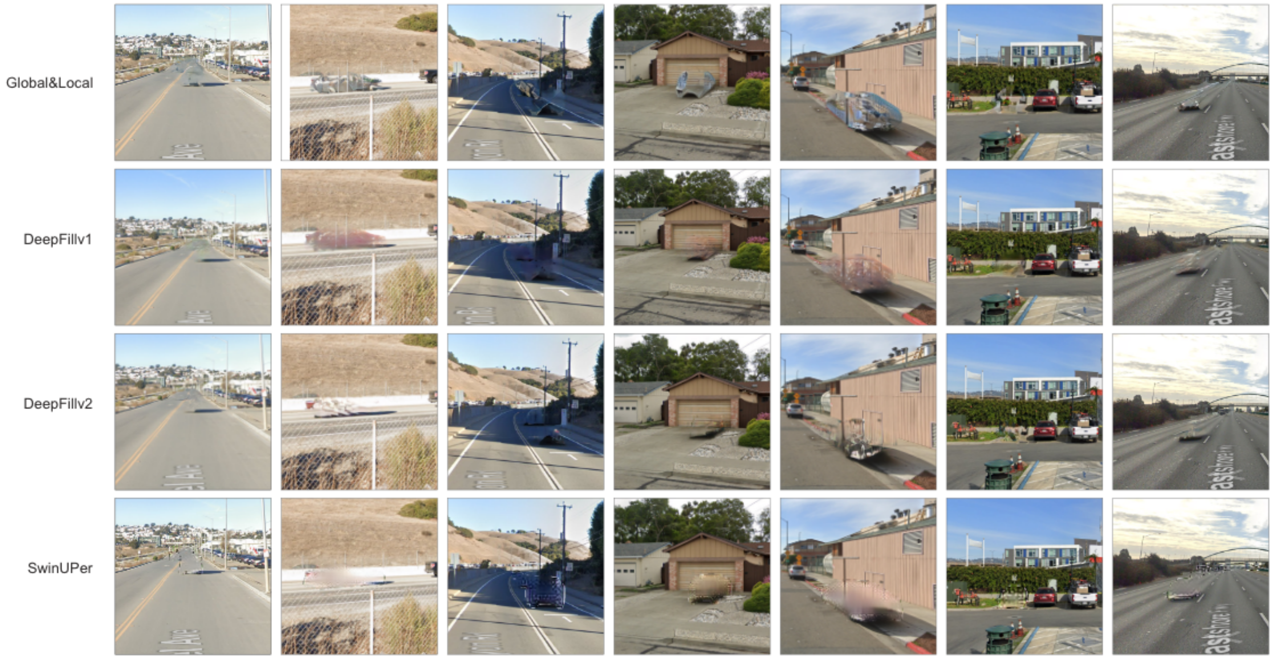


Figure 4. Inpainting results from Global&local, Deepfilv1, Deepfilv2, and SwinUPer

that reduces the inpainting quality. We chose a dilation kernel of 7 to make sure that our mask covers the edge of the selected object.

4.2. SwinUPer

Our proposed inpaintor, SwinUPer, uses a one-stage generator and a spectral-normalized discriminator. For our SwinUPer generator, we used a ImageNet-22k pretrained Swin Transformer as our encoder for the image. Pretrained Swin Transformer contains 4 layers, and it outputs all 4 different scales of feature maps. We uses a 4-layer UPerNet as our decoder to produce our inpainted image. UPerNet's feature pyramid combines different depth of attention fields from Swin Transformer, and serves to provide global and local attention for our image completion.

We adapt SN-PatchGAN loss proposed by [18] for training. Our loss for generator is hinge loss + l1 loss for hole + l1 loss for the rest of the image, and SN-GAN loss for the discriminator:

$$L_{hinge} = -\mathbb{E}_z[D^{sn}(G(z))]$$

$$L_{l1} = \sum_{i,j\in \text{mask}} \|x(i,j)-z(i,j)\| + \sum_{i,j\notin \text{mask}} \|x(i,j)-z(i,j)\|$$

$$L_G = L_{hinge} + L_{l1}$$

$$L_{D^{sn}} = \mathbb{E}_x[ReLU(\mathbb{1}-D^{sn}(x))]+\mathbb{E}_z[ReLU(\mathbb{1}-D^{sn}(G(z)))]$$

where z is the masked image, x is the label image, i, j are pixel coordinate, mask is the subset of pixels that are masked.

5. Results

We trained our Transformer inpaintor pipeline end-to-end for 50000 training steps on 8 Nvidia Tesla K80s for 3 days on Places365 training dataset. The evaluation inference speed is at 50ms per image for images of resolution 256 × 256, with uniformly random hole area of 15%∼65%.

5.1. Quantitative Results

We evaluated our inpaintor against mmediting[1]'s implementation of GlobalLocal[6], DeepFillv1[17] and DeepFillv2[18] on Places365 validation set. Input masks are free-form for all except GlobalLocal, which is evaluated on bounding box masks. The results are summarized in the following table:

Model	l1-loss	PSNR	SSIM
Global&Local	0.256	21.84	0.849
DeepFillv1	0.250	22.19	**0.850**
DeepFillv2	0.184	21.79	0.806
SwinUPer	**0.171**	**22.69**	0.807

Table 1. Evaluation Results on Places365 Validation Set.

From l1-loss and PSNR, SwinUPer outperforms all previous models.

5.2. Qualitative Comparisons

In Fig.4, we compare the qualitative predictions from GlobalLocal[6], DeepFillv1[17], DeepFillv2[18], and

SwinUPer in removing cars from sample Google Street Views pictures using our pipeline.

SwinUPer seems to be performing very badly because it outlines the mask with black and white artifacts. However, at a further-away look without considering the edge of the mask, SwinUPer completes the image very well. We notice that there is a pattern in the edge artifacts - the completed mask is often square-like, leaving the rounded edges white and black. For example, on column 3, SwinUPer completes only the square body of the car. We believe that the edge artifacts of SwinUPer is caused by Swin Transformer's windowing methods. Furthermore, the SwinUPer predictions, at a closer look, also has an interleaving block-like pattern. This could be caused by insufficient structural information of the image, since it is only one-stage.

6. Discussion

There are a few changes in our pipeline that may lead to higher performance. First, if we use negative infinity instead of 0 to represent our mask, this would lead to the inputs of the second layer's embeddings also masked in the same way as the original mask, since apply softmax on infinity would result in 0. The choice of mask representation can be explored. Second, because SwinUPer always outputs a blurry mask completion, it is possible that it lacks structural understanding of the image, which can be improved by a second stage of contextual attention. Third, the architecture of SwinUPer could be updated for free-form inpainting needs, since windowing has created square-like artifacts in our predictions. Lastly, the shadow of the removed object still residue in the inpainted image. This could be mitigated by training a segmentation method which aims to output a mask including its shadow.

7. Conclusion

We developed a DR-ready pipeline using state-of-the-art inpainting methods, and also propose SwinUPer, a novel inpainting model that could become the state-of-the-art for inpainting. Despite the lack of compute resources, a partially trained SwinUPer inpaintor has already shown its potential in achieving a new state-of-the-art for inpainting. For our next step, we hope to address issues mentioned in discussion, and also explore the possibility of using our existing inpainting pipeline as a self-supervised pretraining task.

References

[1] MMEditing Contributors. Openmmlab editing estimation toolbox and benchmark. `https://github.com/open-mmlab/mmediting`, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[6] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

[7] Norihiko Kawai, Tomokazu Sato, Yuta Nakashima, and Naokazu Yokoya. Augmented reality marker hiding with texture deformation. *IEEE Transactions on Visualization and Computer Graphics*, 23(10):2288–2300, 2017.

[8] Norihiko Kawai, Tomokazu Sato, and Naokazu Yokoya. Diminished reality based on image inpainting considering background geometry. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1236–1247, 2016.

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[11] Shohei Mori, Sei Ikeda, and Hideo Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–14, 2017.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[14] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[16] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[17] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 5505–5514, 2018.

[18] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[19] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.