SEGMENTATION GUIDED IMAGE INPAINTING

A Thesis

by

STUTI SAKHI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Nima Kalantari
Committee Members,	Zhangyang Wang
	Xiaoning Qian
Head of Department,	Scott Schaefer

August 2020

Major Subject: Computer Science

ABSTRACT

Deep Learning based approaches have shown promising results for the task of image inpainting. These methods have been successful in generating semantically correct and plausible inpainted images. In case of object removal, these methods require the input image to be masked roughly around the object region. The process of masking the input image causes loss of useful information as background pixels are also masked out by the rough mask. This loss of useful information makes the inpainting networks highly dependent on the mask shapes and size. The quality of the inpainted image deteriorates as the mask size increases. In our work, we propose a segmentation guided inpainting network which is not dependent on the mask shape and size for object removal. It learns to classify the foreground and background spatial locations in the mask region and uses them accordingly for the image reconstruction. This network takes the complete image as input along with the mask as a separate channel and outputs the inpainted image with the object removed. We also generate a paired dataset of image with the object and without the object which is required to train this fully supervised network.

DEDICATION

To my parents, Anil Sakhi and Savita Sakhi, this work would never have been possible without you. Thanks a lot for making me the person I am today.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my committee chair, Dr. Nima Kalantari for his constant support and guidance throughout the course of this research. I would like to thank my committee members, Dr. Zhangyang Wang, and Dr. Xiaoning Qian, for their suggestions and evaluable feedback of this work. Thanks also go to my lab mates, friends and the department faculty and staff for making my time at Texas A&M University a great experience. Finally, thanks to my mother, father, sister and grandmother for their encouragement, love and constant support.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Dr. Nima Kalantari and Dr.Zhangyang Wang of the Department of Computer Science and Dr. Xiaoning Qian of the Department of Electrical and Computer Engineering. All the work conducted for the thesis was completed by the student independently.

Funding Sources

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Image Inpainting, the task of restoring old and damaged images has been around for a really long time. In the recent years, image inpainting has gained huge popularity in the domain of digital images due to the development of various advanced image processing techniques. More specifically in the digital domain, image inpainting is the task of reconstructing missing or corrupted pixels of an image while maintaining its structural and textural coherency. Although restoring images, removing objects and text from images are its obvious applications, image inpainting finds applications in various tasks including image-based rendering, super resolution, image stitching, compression and many others.

In the past, various diffusion based and patch based algorithms [1, 2, 3, 4] have been proposed to solve the problem of inpainting. These methods work by interpolating the neighboring pixels into the hole region or by replacing the hole region by the best fitting patch found in the rest of the image. The efficiency of these methods is limited to small, narrow whole regions and images with a global texture. These methods do not hold the ability to utilize semantic information available in the image and hence fail to perform in images with high structural complexity.

Recently, deep learning methods have shown huge potential in learning semantic features of an image. Various deep learning networks have been proposed for the task of image inpainting as well. [12] is the first deep learning framework that was proposed to perform

image inpainting for rectangular hole regions. Later, this method was extended by [8]. They proposed an architecture which allowed for better semantic feature extraction and also aimed at performing inpainting for irregular holes. To further improve the performance of inpainting in irregular holes, [12] and [23] proposed novel convolution operations which took the mask in consideration while performing the convolution operation. These methods perform considerably well in structured cases and can even hallucinate objects to fill in the missing area.

Very recently, efforts have been made to generate more detailed inpainting results with sharper boundaries for various regions in the hole. These methods explicitly incorporate the learning of various structural information - contour [20], semantic map [17], edge [11], into the network to aid the process of inpainting. These are two step networks where the first network predicts the structural information of the hole region which is utilized by the second network which performs the actual inpainting task.

All these proposed methods treat the task of object removal similar to the task of hole filling. They perform object removal from an image by first masking the image with a rough mask around the object and then fill in this masked area using the network. We need to note that apart from the pixels belonging to the object (to be removed), the process of masking the image with a rough mask also removes some pixels which do not belong to the object.

The background pixels within mask region, if not removed, can be used in the inpainting process to generate more detailed and semantically correct images. These removed background pixels can be of much more importance in cases of highly structured images where generating sharp boundaries is a challenge. In most cases, the quality of inpainted image especially in terms of semantic correctness deteriorates with the increase in size of the mask.

In this work, we propose a segmentation guided inpainting network which incorporates the usage of all the background pixels to specifically solve the problem of object removal from an image. The proposed network is a three- step network that removes an object from an image by segmenting the object to be removed along with inpainting the region belonging to the object in the image. The network takes the complete image (as opposed to masked) and rough mask as input in separate channels and generates both the segmentation mask of the object to be removed and the inpainted image as output. Taking the complete image as input unlike the masked image in previous approaches, enables least information loss and allows the network to learn identifying the spatial locations which are necessary for reconstruction and use them accordingly. The network is explicitly made to output the segmentation mask of the object to be removed, to aid this classification of foreground and background spatial locations by the network.

To train this fully supervised inpainting network which focuses on object removal, we require a dataset which contains paired images- with and without the object of interest.

However, such a dataset with a size comparable to other commonly used inpainting datasets [24], is not publicly available. So, we create our own paired dataset using Places2 [24] and COCO [9] dataset which are both publicly available.

With this work, we have two major contributions. Firstly, we propose a three- step network for object removal which incorporates the use of all background pixels in the generation of inpainted output. Secondly, we create a paired image dataset – with and without an object, to train a fully supervised object removal network. We compare the performance of our proposed network with other state of the art inpainting networks for object removal. We also show how the output inpainted image remains almost unaffected by increase in object size, thus removing the dependency on mask shape and size.

2. RELATED WORKS

Numerous Image Inpainting techniques have been proposed in the past. We discuss the development of these inpainting algorithms in the section below.

Diffusion based methods propagate the neighboring pixels into the hole region for interpolation. They start from the boundaries of the hole and move inwards filling the region. The method proposed by [1], enforces constraint of isophate lines arriving at the boundaries to be completed in the hole region to propagate the pixels in the hole. However, the reconstruction is dependent only on local pixels and thus fails to be globally coherent. Diffusion based methods are also effective only in cases of small and narrow holes.

Patch based methods fill in the missing pixels by a patch in the undamaged portion of the image. These algorithms scan through the image patch by patch looking for the best match for the hole region. [3] and [4] proposed non parametric patch patching algorithms which worked by assuming a markov random field and building the hole region pixel by pixel by finding all similar neighborhoods. However, these methods had high memory and computation requirements. To reduce these requirements a randomized algorithm, [2] was developed. Though the patch based methods perform well in a consistently textured image, it fails to perform in cases where the texture is unique to the hole region. It also fails to semantically pleasing results in highly structured images.

Recently, various deep learning methods have been proposed to perform automatic inpainting. Neural Networks have the ability to learn the semantic features of an image required for inpainting and thus perform better than the traditional methods in relatively structured scenes. Context Encoder [12] is one of the first deep learning based methods for image in which uses an encoder-decoder architecture to perform inpainting in rectangular hole regions. They explore the adversarial loss [6] along with a standard pixel wise reconstruction loss for training which helped produce sharper images as opposed to using just the reconstruction loss. Although the method generated semantically plausible results, the filled regions lacked the textural details and the network was constrained to take only rectangular hole regions as input.

[8] built upon Context Encoders and proposed a fully convolutional inpainting network to fill arbitrary shaped holes in high resolution images. The network is trained using local and global discriminators to allow the generated images to maintain both local and global coherency. They also employed dilated convolutions in all the layers of the generator allowing for a greater receptive field without increasing the number of learnable parameters to improve global coherency. Fast marching method [18], followed by Poisson image blending [13], is employed as a post processing step to remove the color inconsistencies in the hole region and surrounding areas. The method produces visually pleasing results with reasonable semantic correctness and textural details but still relies on a post processing step to perform color corrections and is not free of artifacts.

Convolutional neural networks are not particularly effective in learning long distance correlations thus creating boundary artifacts and distorted structures in the inpainted results. [22] proposed a two-step feed forward generative network with a novel contextual attention layer. This contextual attention layer allows for the information distant from the hole to be available for use in the hole filling. They use a local as well as global discriminator and train the network adversarially using the [7] loss. They also employ a spatially discounting reconstruction loss to allow for higher freedom for hallucination of pixels by weighing the reconstruction loss higher at the boundary of the hole when compared to the regions away from the boundary.

Vanilla convolution filters treat both the valid and hole pixels in the input image equally and hence the extracted features depend on the hole pixel values as well. The dependency on the initial hole values is attributed to several issues in the inpainted results like color contrasts and edge artifacts. [10] proposed a novel partial convolution layer to address this dependency on the initial hole values in the input image. Partial Convolution performs masked convolution and renormalizes the output to condition only on the valid pixel. The convolution is followed by a mask update step. This network demonstrated the efficacy of training image-inpainting models on irregularly shaped holes.

Partial convolution heuristically categorizes pixel locations to be valid or invalid and thus multiplies hard gating values to the input feature maps. Moreover, invalid pixels disappear in deep layers making all the gating values to be 1. To tackle this problem [23] proposed

a gated convolution which allows the network to learn the optimal gating values. This allows for a dynamic feature selection mechanism for each channel, at each spatial location, across all layers. They use a two-step network and also propose a novel GAN discriminator SN-PatchGAN which eliminates the need to have both local and global discriminators. Gated Convolution effectively eliminates the color inconsistencies and generates visually pleasing results.

The previously mentioned deep generative models enabled an efficient end-to-end framework for image inpainting, but these methods don't exploit image structure knowledge explicitly to constrain the object shapes and contours, which usually lead to blurry results on the boundary and color bleeding to other regions. Recently various networks have been proposed which utilize explicit image structure knowledge for inpainting to generate better object boundaries. [17] proposes SPG-net which first predicts the segmentation labels in the missing area and then generates inpainting results utilizing the predicted segmentation labels. They use state of the art segmentation networks to initialize the segmentation labels and train a network to predict segmentation labels in the hole region. This predicted semantic label map along with the incomplete image are input to the inpainting network which outputs the inpainted image. [11] proposes another approach of inpainting by making the network explicitly learn the edges of the missing region, thus allowing for sharper and cleaner boundaries. The edge generator hallucinates edges of the missing region of the image, and the image completion network fills in the missing regions using hallucinated edges as a priori. [20] also followed the same strategy

and proposed a foreground aware inpainting system which cleanly separates out the contour prediction task from image completion. The contour for the hole region is predicted which is used in the image inpainting to generate better boundaries and sharper images.

3. PROPOSED APPROACH

We consider the process of object removal given the complete image and a mask as a three step process. Firstly, the spatial locations within the mask region which belong to the object are identified i.e., the process of segmenting the object of interest present in the mask region. Making use of this segmentation mask, a coarse inpainted image is generated in the second step. The goal of this process to grab the regions within the mask which belong to the background and use them directly in the output image. The object region is coarsely filled using all the replicated background pixels. This step also plays a major role in eliminating any minor errors in the segmentation process. The last step takes the segmentation mask, input mask and coarse image and fills in the details to the image. This results in the final output inpainted image with the object removed. To replicate this three-step process we propose a three-step network for object removal. The 3 components are namely – segmentation network, coarse network and fine network. Figure 3.1 shows the pipeline of our proposed network. The strategy of using two step networks for inpainting has been adopted from [22]. In the subsections to follow, we discuss each of the networks in detail.

3.1. Segmentation Network

Segmentation Network takes image and random mask as input. It predicts the segmentation mask of the object within the mask region as output. Segmentation Network follows encoder-decoder architecture adopted from UNet [14]. The task of segmentation

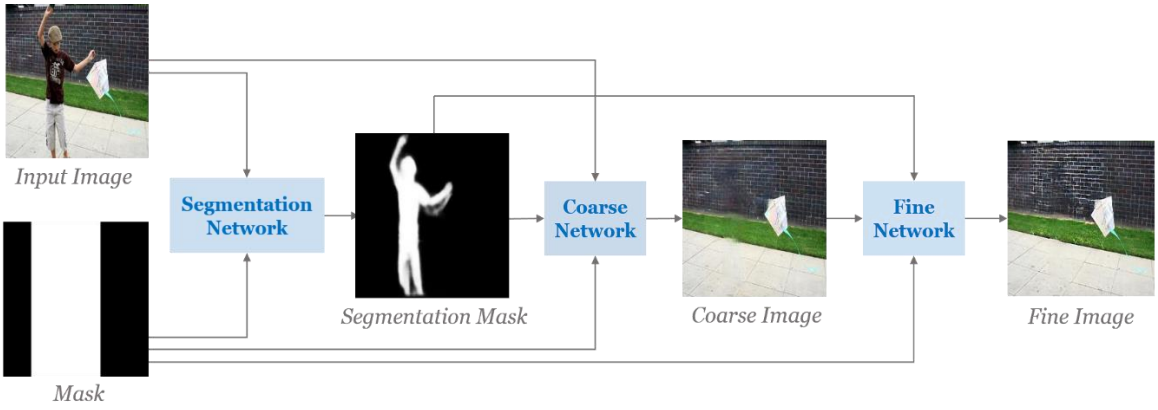

Figure 3.1 Proposed Three-Step Network Pipeline

is particularly difficult as a prediction needs to be made for every spatial location. Skip connections used in the Unet architecture allow for better reconstruction in the decoder part of the network. We however, have larger sized kernels in the starting layers of the encoder part unlike Unet to provide a higher receptive field which is essential in the segmentation process. Figure 3.2 shows the architecture of Segmentation Network in detail.

In our network, segmentation is a pixel wise two-class classification task where each pixel is classified as foreground (object) or background. We use binary cross entropy loss to train the segmentation network. Equation 3.1 represents the segmentation loss function. X and T are predicted and true segmentation masks respectively. N is total number of pixel in the segmentation mask. t_i and x_i represent the ith pixel in T and X respectively.

$$BCE(X, T) = \sum_{i=1}^{N}(t_i \log x_i + (1 - t_i) \log(1 - x_i)) \quad (3.1)$$

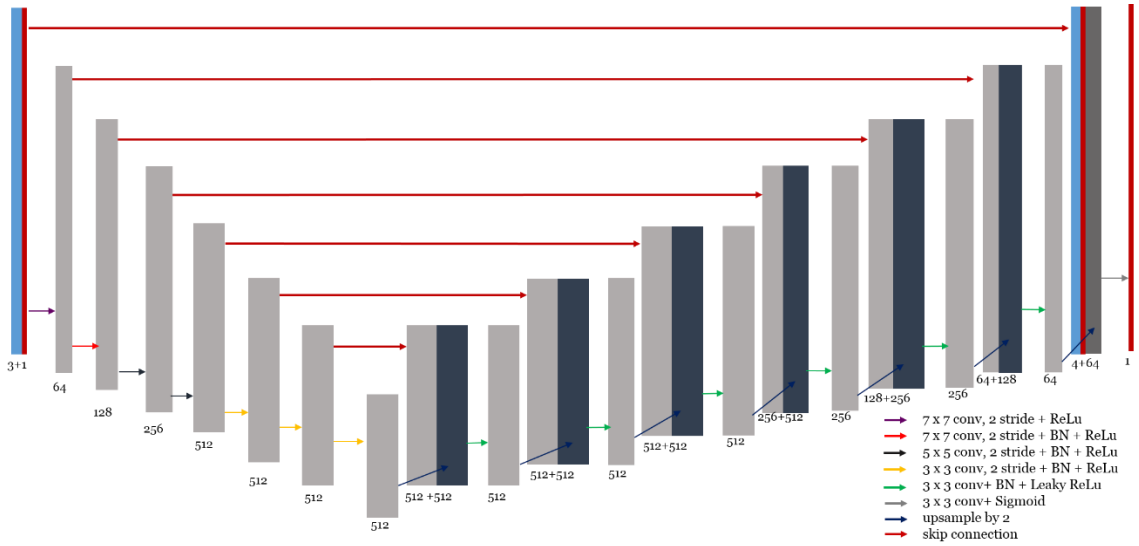

Figure 3.2 Segmentation Network Architecture

3.2. Coarse Network

Coarse Network takes the segmentation mask outputted by the segmentation network along with the input image and mask as inputs and outputs a coarsely inpainted image. Coarse Network tries to grab all possible background pixels from the input image and uses these pixels to coarsely fill the object region. The network architecture is adopted from [23]. It is an encoder-decoder architecture with a series of dilated convolutions in the center. The task of inpainting requires both global and local semantic understanding. The dilated convolutions provide a large receptive field which aids global semantic understanding. Figure 3.3 shows the architecture of Coarse Network in details.

We use weighted L1 reconstruction loss to train the coarse network. Higher weight is given to the pixels within the input mask. Equation 3.2 represents the reconstruction loss function.

$$L_{Recon} = \alpha \left\| M * (I_c - I) \right\|_1 + \beta \left\| (1 - M) * (I_c - I) \right\|_1 \tag{3.2}$$

I and M are the input image and input mask respectively. I_c is the coarse image output by the coarse network. α and β are the weights for masked region and unmasked region respectively. The values of α and β are taken as 3.0 and 1.0.

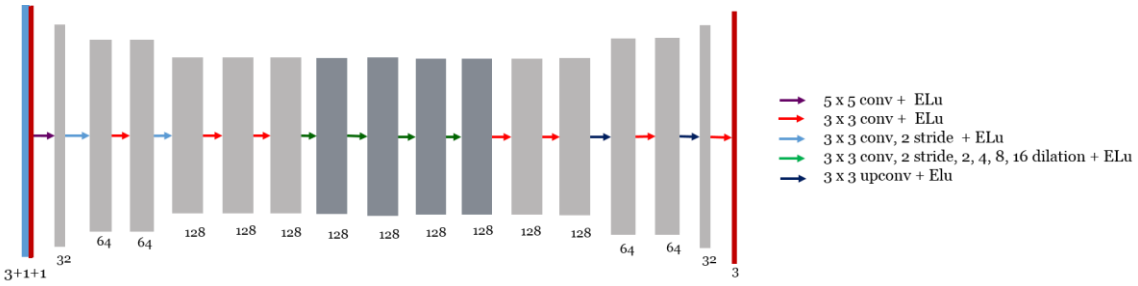

Figure 3.3 Coarse Network Architecture

3.3. Fine Network

The Fine Network is the last network in the three-step network. It takes in the coarse image, segmentation mask and input mask as inputs and outputs the final detailed inpainted output. It performs the job of adding further details to the coarsely filled regions in the coarse image. The architecture for this network has been adopted from [23] which has two branches – dilated convolution and contextual attention. The dilated convolution

branch works similar to coarse network and allows for a semantic understanding at global level. The contextual attention branch on the other hand allows the network to learn from where to borrow information from the background patches to fill in the missing patches. Figure 3.4 shows the detailed architecture of the Fine Network.

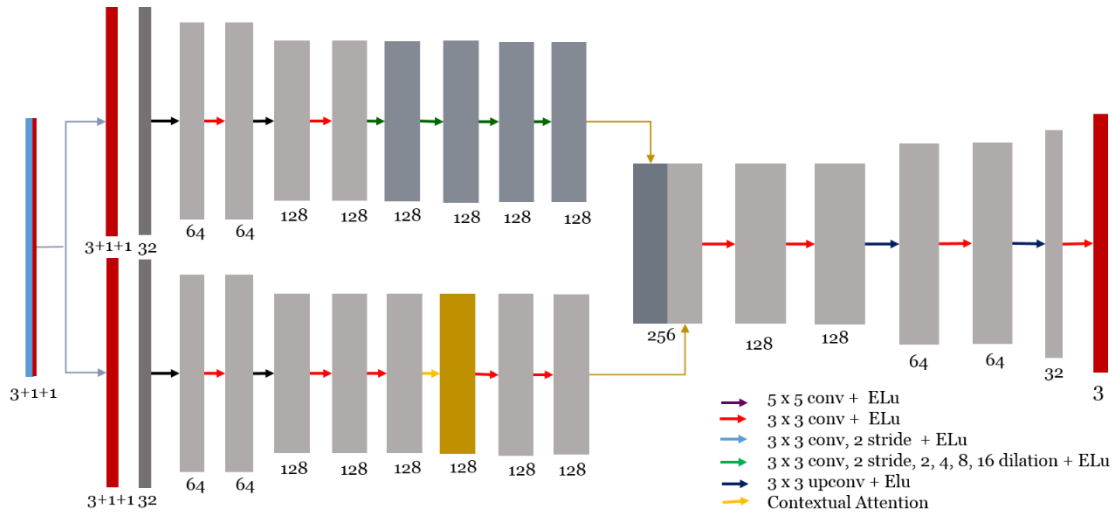

Figure 3.4 Fine Network Architecture

A weighted sum of two losses is used to train the fine network. Firstly, L1 loss is used to provide the basic guideline for reconstruction. Equation 3.3 represents the L1 loss. I_f is the predicted fine image and I is the ground truth image.

$$L_{Recon} = \|(I_f - I)\|_1 \tag{3.3}$$

Secondly, GAN loss is used which is essential for adding details by hallucinating textures as well as replicating them. We use the Spectrally Normalized PatchGAN proposed in [23] which has been widely used for image inpainting task since. We also adopt the hinge GAN loss as proposed in [23]. Equations 3.4 and 3.5 represent the generator hinge GAN loss and discriminator hinge GAN loss respectively.

$$L_{GAN_{gen}} = -E_{pred}\left(D\left(G\left(I_c,\ M,\ M_{pred}\right)\right)\right) \tag{3.4}$$

$$L_{GAN_{dis}} = E\left(ReLu\left(1 - D(I)\right)\right) + E_{pred}\left(ReLu\left(1 + D\left(G\left(X,\ M,\ M_{pred}\right)\right)\right)\right) \tag{3.5}$$

G is the generator network i.e., the fine network. D is the SN Patch GAN. M and M_{pred} are the input mask and predicted segmentation mask respectively.

4. TRAINING PROCESS

4.1. Dataset

Training the fully supervised network we proposed, requires a dataset which contains paired images - with and without the object of interest. Due to unavailability of such a dataset of considerable size we create our own dataset. In this section, we discuss the process we followed to generate this dataset. To be specific, we require a dataset which provides us the following - image with object of interest, image without the object of interest and the segmentation mask of the object.

We use two publicly available datasets to generate our required dataset - Places2 [24] and COCO [9]. Places2 dataset contains more than 10 million images comprising more than 400 unique scene categories. This dataset is commonly used in image inpainting tasks due to large size and varied scene categories. COCO is a large-scale object detection, segmentation, and captioning dataset. COCO dataset provides 80 thousand images comprising 91 object categories along with the respective segmentation masks.

We use images from Places2 dataset as our base image and paste the objects from coco dataset on these images at random spots. We resize the objects covering more than 50% of the image to fit within half of the image to allow reasonable hole sizes. Segmentation masks are also resized and translated in the same way as the object. Also, we filter out objects which cover less than 5% of the image area before pasting them. We also employ

a set of augmentation techniques while pasting the objects. For a given background image, we select a random object image from the COCO dataset. . The object is placed at a random location in the background image after random rotation and random resizing. This allows for generation of a diverse paired dataset for supervised learning along with the segmentation masks. Figure 4.1 shows some sample images from the generated dataset.

Figure 4.1 Generated Dataset – Sample Images

4.2. Training Details

In this section, we specify the training details of our model. Specifically, we train the three networks separately for a number of reasons. Firstly, segmentation network is trained on real images from COCO dataset instead of using the generated datasets. This prevents the network from overfitting to the generated dataset where objects are just superimposed on background images. Both the inpainting networks, coarse and fine are trained using the generated dataset. We train both these networks separately to allow for better object removal. We discuss more on this in the section 6.

All the three networks were trained using the Adam optimizer with a learning rate of 0.0001. The segmentation network and coarse networks were trained for 150K iterations each. On the other hand, fine network was trained for 350K iterations. Fine network was be trained for more iterations as it did not over fit easily. Segmentation Network and Coarse Network have higher chances of overfitting as they have COCO objects in their inputs which are limited in number (140K). The fine network never gets to see these objects as the coarse network already removes the object from the image.

5. RESULTS

5.1. Evaluation

We present an extensive qualitative inspection which is the best way to evaluate the task of image inpainting. Image inpainting lacks good quantitative metrics as a given input can have multiple plausible output inpainted image. We, however, also report the L1 Error and L2 Error to compare our model with previous approaches which report the same metrics. L1 error and L2 error correspond to mean pixel wise L1 and L2 distance between output image and ground truth image.

5.2. Comparison with State of the Art Networks

We present both qualitative and quantitative comparisons with three previous state of the art networks for image inpainting – [10], [22] and [23]. Due to unavailability of official network for [10], we train the network ourselves using rectangular masks for this comparison. For qualitative evaluation, we use COCO validation images and remove the object of interest from them. Rectangular masks are generated from the segmentation masks of the objects available in the COCO dataset. The masked images along with the mask are given as input to the previous state of the art networks. As opposed to the masked image, our network takes the complete image as input. Our network, makes use of the background pixels in the mask region and produces semantically plausible images. We observe three different scenarios in particular where our network performs better than the previous approaches. Firstly, our network generates sharper and semantically correct

boundaries in the masked region. This is a major concern for previous approaches when small portion of background objects are covered by the mask. Figure 5.1 shows the comparison of our network with previous approaches generating semantically correct object boundaries is a major challenge.

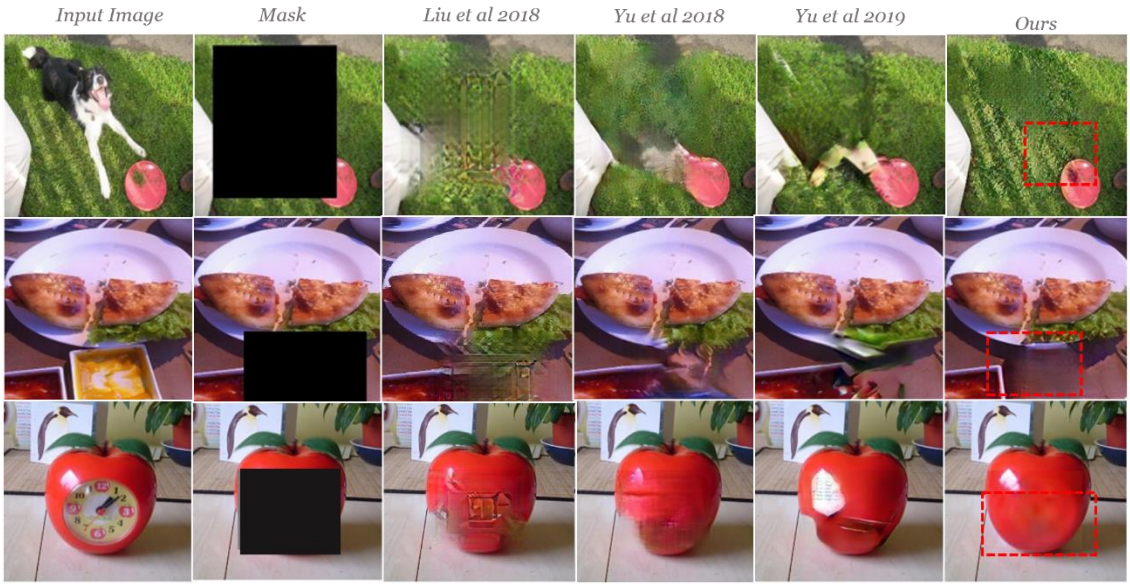

Figure 5.1 Comparison with State of the Art networks – Boundaries

Secondly, our network performs better in cluttered scenes which are particularly challenging for other inpainting approaches. Most inpainting networks fill unrealistic structures in cluttered scenes as they are unable to understand the image semantically. These unrealistic structures, though look blend in at first glance, can be pointed out clearly on a little more observation. Figure 5.2 shows how our network generates realistic completions when compared to previous approaches when it comes to cluttered scenes.

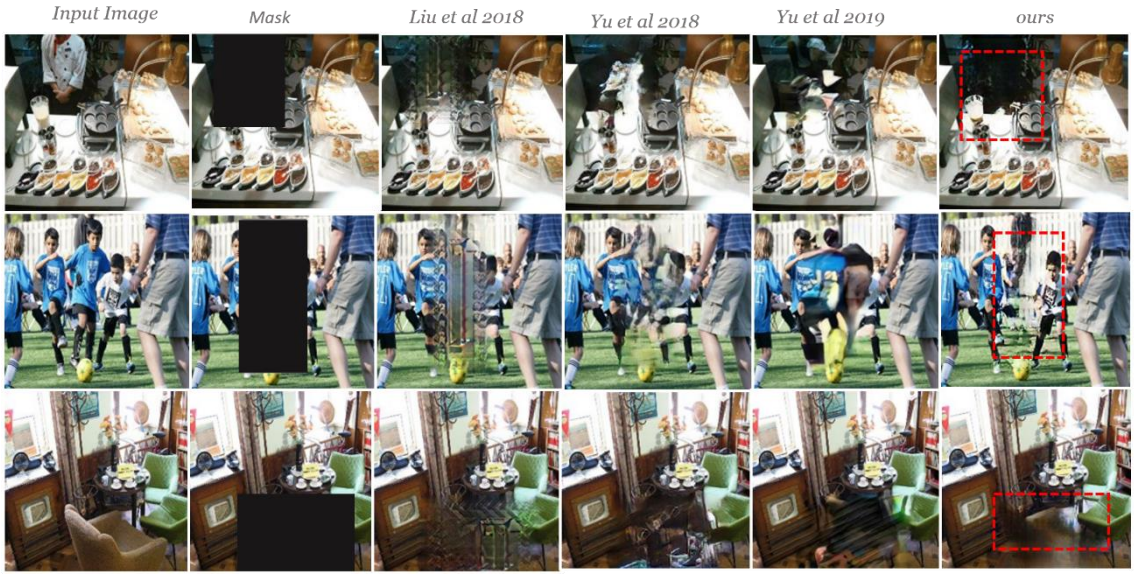

Figure 5.2 Comparison with State of the Art networks – Cluttered Scenes

Lastly, our network performs better in cases where the mask is at edges. Inpainting at edges is more challenging as the available neighboring pixels is drastically reduced. Figure 5.3 shows comparisons for some cases with mask at the edges.

Method	L1	L2
Liu et al. 2018	0.8763	0.5629
Yu et al. 2018	0.8261	0.5355
Yu et al. 2019	0.7846	0.4723
Ours	**0.5229**	**0.2539**

Table 5.1 Comparison with State of the Art Network – Quantitative Metrics

For the quantitative evaluation, paired images are required – with object and without the object. Due to unavailability of such a database, we provide the quantitative evaluation on the generated data using validation images from Places2 and COCO. Table 5.1 shows the quantitative comparison of our network with previous approaches. Our network clearly has lower L1 and L2 error as our network is designed to use and grab as much information possible from the input image.

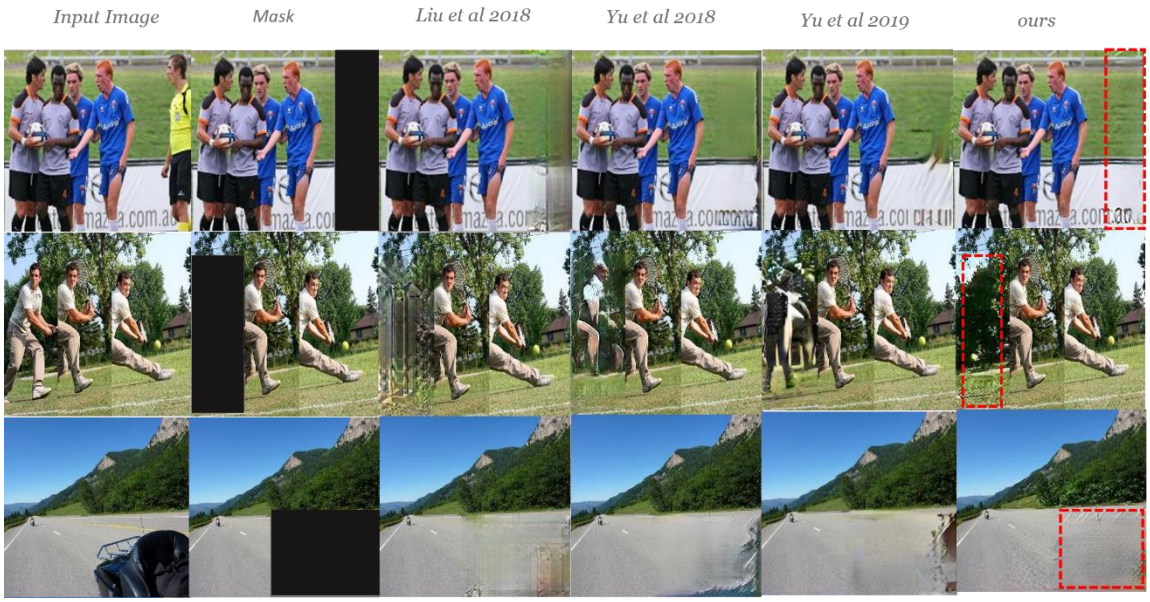

Figure 5.3 Comparison with State of the Art Networks – Edges

6. ANALYSIS

6.1. Performance on Out of Class Objects

Our network is specifically trained for object removal using the objects from images in the COCO dataset. COCO dataset contains images with about 91 unique object categories which cover various broad categories including – person, animals, automobile, appliances, food items and other miscellaneous objects. Though these categories cover most common object categories, the task of object removal might require removal of objects outside these categories, like – flowers, fish, distant building, pathways etc. We qualitatively evaluate the performance of the network on out of class object removal from images. We observe that our network performs equally well with out of class objects. This can be attributed to the generalized nature of segmentation network. Firstly, we made sure to train the segmentation network on COCO dataset and not the generated dataset. Secondly, the mask input given to the network allows the network to generalize and just learn to predict any object inside the mask irrespective of the object class. Figure 6.1 shows some inpainted results on out of class objects. It can be seen that the network performs effectively for all these varied out of class objects.

Figure 6.1 Performance on Out of Class Objects

6.2. Dependency on Mask Shape

One of the major motivations of this work is to remove the dependency of the object removal process on the input mask shape and size. Our network is especially designed to get rid of this dependency as it inherently classifies the background pixels and foreground pixels and thus not requiring the input image to be masked. Firstly, our network does not require a free form mask like previous methods to perform efficient object removal. Moreover, an increase in the size of rectangular mask also does not affect the quality of output. Table 6.1 show the quantitative evaluation for the same. We have used three different mask sizes based on margins in pixels – 5pi, 30pi, 60pi, for the quantitative evaluation. As it can be seen, the evaluation metrics remain almost the same over all mask

sizes. Qualitative evaluation for the same can be seen in Figure 6.2. It shows how the results of the inpainting remain almost the same with increasing mask shapes.

Mask Margin	L1	L2
5pi	0.49	0.22
30pi	0.46	0.19
60pi	0.50	0.21

Table 6.1 Dependency on Mask Size – Quantitative Metrics

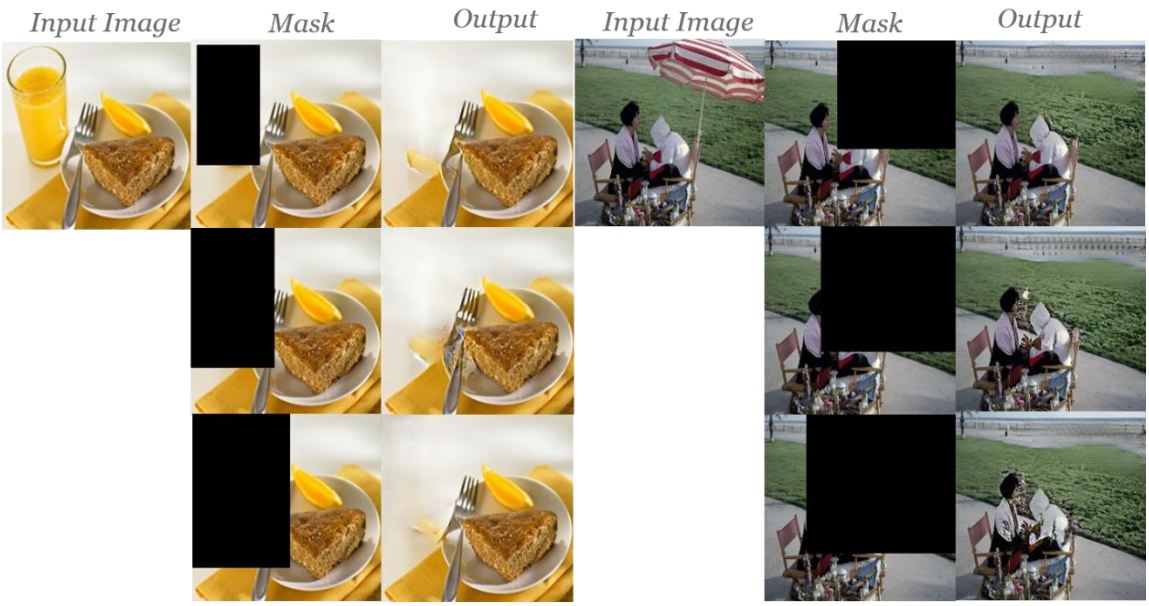

Figure 6.2 Dependency on Mask Shape – Qualitative Evaluation

6.3. Segmentation Error Propagation

One of the major drawbacks of serially connected networks is the propagation of error from one network to another. This is one of the challenges for our network as the coarse network depends highly on the segmentation network to identify the spatial locations which need to be discarded. In cases where the segmentation mask fails to predict a precise masks, coarse network is prone to make errors by leaving small portions of objects in the output image. Our network, however, is resistant to errors in segmentation masks especially for cases when the mask lacks exact boundaries. On the other hand, if the same predicted mask is used to mask the images for input to previous approaches, they result in artifacts in the images due to lack exact boundaries. It is impossible to eliminate the propagation of error when using previous approaches. Our network manages the minor flaws in segmentation mask effectively. This property can be attributed to the 2 step nature of the network. We discuss in section 6 about the architecture and training choices which contribute towards lesser segmentation error propagation. Figure 6.3 shows some cases where the network effectively manages the flaws in segmentation mask. It also shows the output of [23] when the same segmentation mask is used to mask the input image.

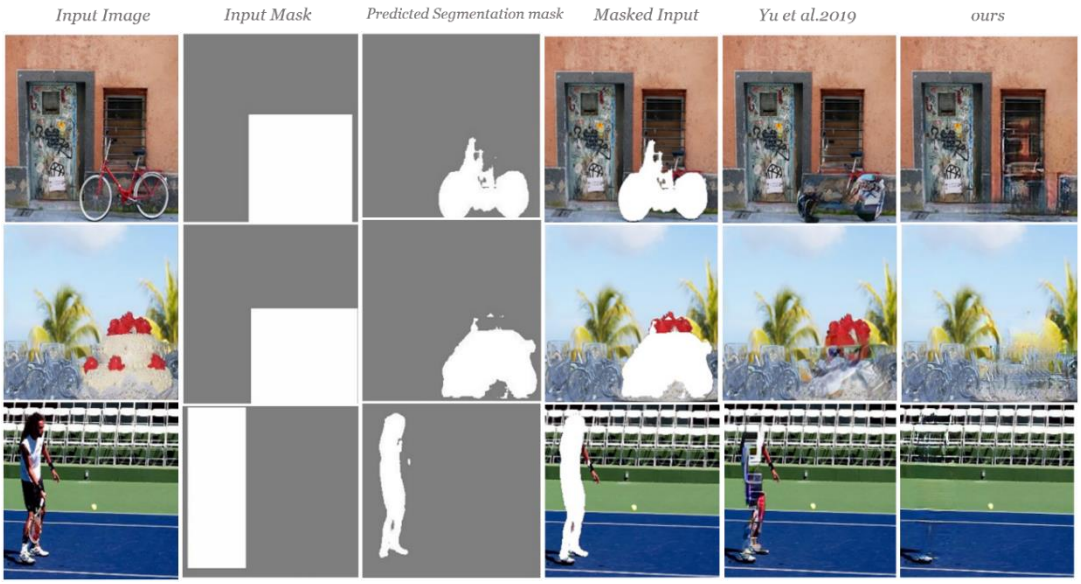

Figure 6.3 Segmentation Error Propagation

6.4. Architecture & Training Choices

One of the architectural choices that we made for the network was to have a two-step network for inpainting or a single step network. Single step networks generate reasonable inpainted images when trained with a GAN loss.

For our approach, we find the use of two step network to be necessary. The two step process allows the network to replicate the step wise process and demarcate the inpainting task effectively. This demarcation allows the coarse network to only focus on identifying the background pixels in the mask region and also effectively manage minor errors in segmentation mask. This proper object removal and coarse filling provides clear guidance to the fine network and also increases its receptive field allowing for accurate detail additions. Single step network on the other hand, failed to remove the object completely

even in cases with almost no segmentation error. Figure 6.4 displays this difference in the performance of object removal for one step vs two step network.

Training strategy for the Two-Step Inpainting network played a crucial role in the object removal process. Training them separately aided the process of object removal and also allowed for more stable GAN training for the fine network. Figure 6.4 displays this difference in the performance of object removal when the networks were trained together vs when they were trained separately.

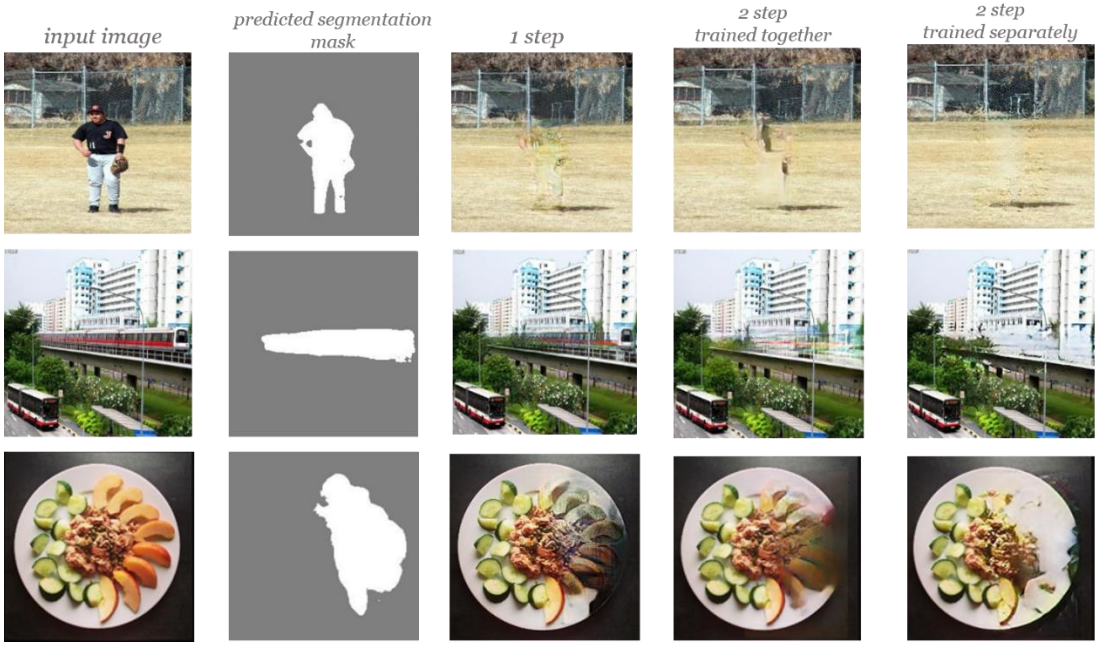

Figure 6.4 Architecture & Training Choices

7. CONCLUSION

7.1. Drawbacks

We have demonstrated the effectiveness of our proposed network under situations with minors errors in segmentation mask as well as in case of out of class objects. The network however still fails to remove the objects completely for particularly difficult cases. These cases include when the object to be removed is occluded or camouflaged. This results in errors in the segmentation mask by a huge margin which the inpainting network cannot manage. Figure 7.1 shows some of these difficult cases where object could not be removed successfully by our network.

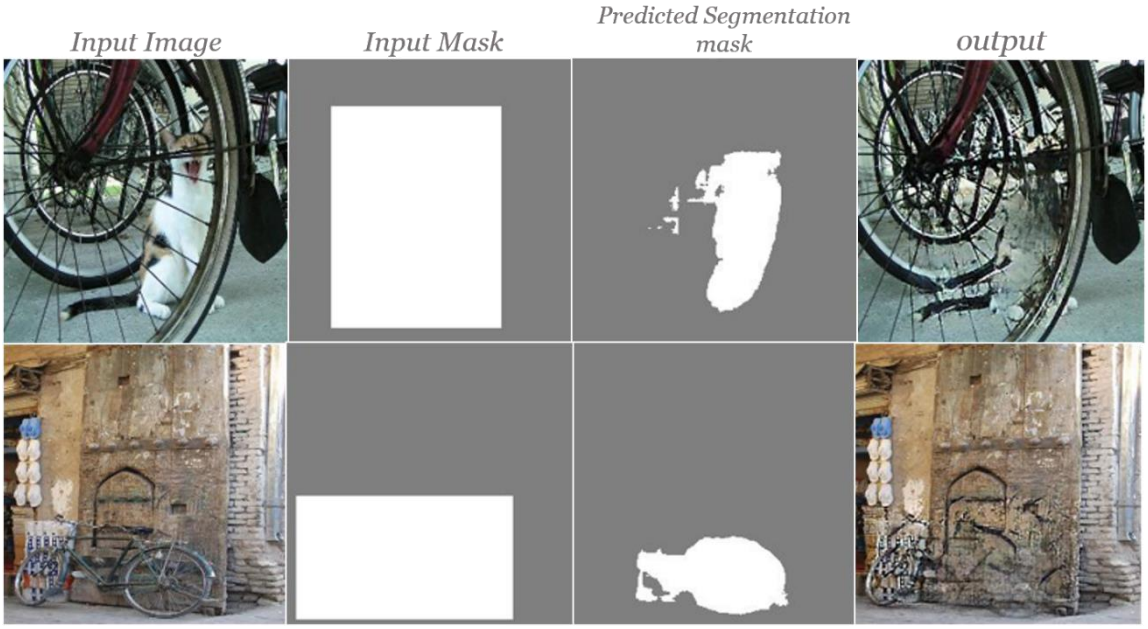

Figure 7.1 Drawback – Propagation of Segmentation Error

The network also fails to identify the shadows and reflections of the objects to be removed. As a result, even though, the object is successfully removed from the image, its shadow or reflection remains in the image. Figure 7.2 shows some examples to demonstrate this drawback.

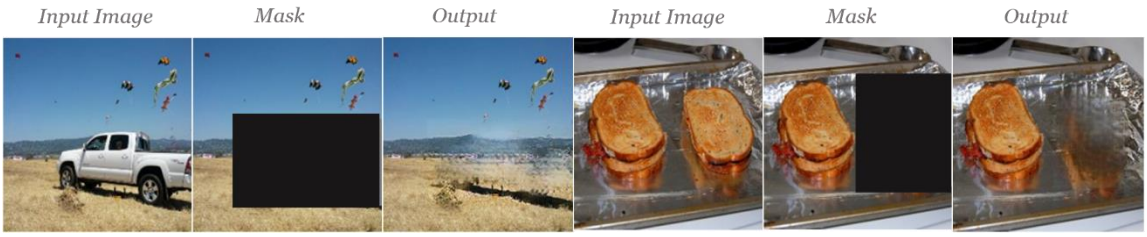

Figure 7.2 Drawback – Shadow & Reflection

7.2. Future Work

Our present work focusses on removing opaque objects which can easily be extended to translucent objects like glass and watermarks. The region within these translucent objects also contain useful information which can be used for efficient reconstruction. Each pixel in an image can be considered to be a weighted sum of foreground and background pixels. This weight is called the alpha value and there are several state of the art networks predicting the alpha mask of an image. We would replace the segmentation network with a matting network which predicts the alpha mask value at each spatial location in the mask region. Changes need to be made to the dataset as well by adding matting datatset to the COCO dataset we are already using. This would result in a more generalized network

which can perform object removal with minimum information loss for any kind of object – opaque and translucent.

7.3. Conclusion

In this work, we propose a segmentation guided three step network for object removal which utilizes all available background pixels for reconstruction in the object region. This network is minimizes information loss irrespective of the mask shape and size. Our network generates sharper and clearer boundaries in the output inpainted image when compared to state of the art inpainting networks. The network generates realistic objects in cluttered scenes and also generates clean results for masks with edges. Moreover, the network manages minor errors in segmentation mask efficiently and removes objects of a wide range of classes effectively.

REFERENCES

[1] Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., & Verdera, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. IEEE transactions on image processing, 10(8), 1200-1211.

[2] Barnes, C., Shechtman, E., Finkelstein, A., & Goldman, D. B. (2009, July). PatchMatch: A randomized correspondence algorithm for structural image editing. In ACM Transactions on Graphics (ToG) (Vol. 28, No. 3, p. 24). ACM.

[3] Efros, A. A., & Freeman, W. T. (2001, August). Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 341-346).

[4] Efros, A. A., & Leung, T. K. (1999, September). Texture synthesis by non-parametric sampling. In Proceedings of the seventh IEEE international conference on computer vision (Vol. 2, pp. 1033-1038). IEEE.

[5] Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In The IEEE conference on computer vision and pattern recognition, 272–280.

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[7] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. In Advances in neural information processing systems (pp. 5767-5777).

[8] Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2017). Globally and locally consistent image completion. ACM Transactions on Graphics (ToG), 36(4), 1-14.

[9] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[10] Liu, G., Reda, F. A., Shih, K. J., Wang, T. C., Tao, A., & Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 85-100).

[11] Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., & Ebrahimi, M. (2019). Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212.

[12] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2536-2544).

[13] Pérez, P., & Gangnet, M. (2003). BLAKE. 2003.“. Poisson image editing.” In Proceedings of ACM SIGGRAPH, 313-318.

[14] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

[15] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.;et al. 2015. Imagenet large scale visual recognition challenge. International journal of computer vision 115(3):211–252

[16] Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[17] Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., & Kuo, C. C. J. (2018). Spg-net: Segmentation prediction and guidance network for image inpainting. arXiv preprint arXiv:1805.03356.

[18] Telea, A. (2004). An image inpainting technique based on the fast marching method. Journal of graphics tools, 9(1), 23-34.

[19] Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8798-8807).

[20] Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., & Luo, J. (2019). Foreground-aware image inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5840-5848).

[21] Yang, J., Qi, Z., & Shi, Y. (2020). Learning to Incorporate Structure Knowledge for Image Inpainting. arXiv preprint arXiv:2002.04170.

[22] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5505-5514).

[23] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2019). Free-form image inpainting with gated convolution. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4471-4480).

[24] Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., & Oliva, A. (2016). Places: An image database for deep scene understanding. arXiv preprint arXiv:1610.02055.