Generating defocused images using deep networks

Priyank Pathak [1], **Prof. Vinay Namboodiri**[2]

[1]Undergraduate 5th year EE and CS dual Major

[2]Department of Computer Science – IIT KANPUR

{ppriyank,vinaypn}@iitk.ac.in

Abstract. *Current applications of deep learning aim at improved generation of images through super-resolution or generation. In this project the aim is to generate defocused images that are conditioned on the input. The aim is to be able to generate defocused portrait mode images automatically. In order to obtain a network that can generate such defocused images, it needs to be trained in a supervised manner, however, datasets are not currently available at scale to train defocused generations. We have created a dataset to train methods for generating such defocused images. Further, we have evaluated several architectures for generating defocused images in this project.*
Keywords: *Defocused image, Pix2Pix.*

1. Introduction

The project is based on the idea of taking the advantage of the struggle that Deep network faces while retaining detail on reconstructed images. Whereas the current smart phone's portrait mode is based on the fact that given a depth map of a normal picture taken from a depth camera, or the secondary lens on the camera when overlapped with the RGB image, creates blurriness which resembles DSLR effect. Figure 1, shows the defocus achieved by the iPhone camera on a 2D image. I have struggled to find Machine learning research papers deploying the use of Deep Nets to replicate such a blur. Possibly due to the absence of requisite dataset. Such a case also raises the possibility of incorporating the use of Depth and Segmentation powers of Deep Networks to create such an effect.

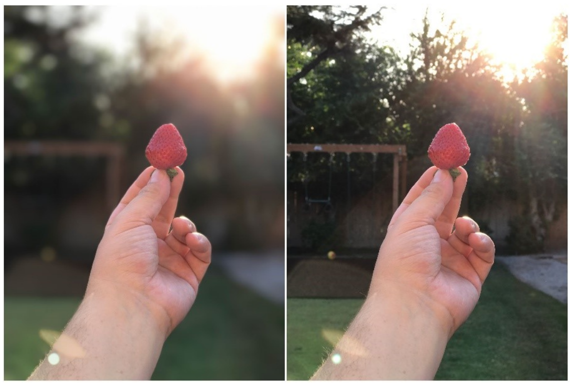

Figure 1. Typical defocused image created from smart phones by overlapping depth map over a RGB photo, picture taken from google[1]

2. Previous Works

As mentioned earlier I haven't found much of the deep learning methodology for creating such blurriness. The closest work are

- The Use of AI to replicate the defocusing in the image, by Adobe. It was announced in summer 2017 sample output were presented in a conference, but no sign of academic paper high lightening their approach [2].
- Facebook demonstrated it's AI capabilities in April'17, where it showcased their photo app which can create blur map on a normal image, yet no sign of research publication yet. [3]
- Research by German scientist Alexander Loktyushin and Stefan Harmeling, Automatic Foreground-Background REFOCUSING, to detect blur and using deconvolution to remove it, hence embarking on the path of superpixel and denoising opposite to our goals. [4]
- IEEE paper by Ruomei Yan and Ling Shao, presents the methodology to detect blurriness and present the segments of image in focus, i.e. a blur assessment technique, hence we can use this as the validation to re-obtain the segmentation of the object of focus, roughly based on low pass filters.[5]
- The closest we have got a paper related to this topic is the paper by University Bayreuth, where they have solved an optimization problem to detect edges and mark the shapes of the object. Use mathematical modeling to mark object and blur accordingly. Results are quite impressive and it perfectly retains the segmentation but results look quite artificial and fails to account for blurring based on distance. Something ensured by Photoshop scripting while applying the gradient blur on the image using gradient depth map. [6]
- This was advised by Namboodiri sir as the close approximation to our problem. Basically, it uses binary pose maps when applied to the image, changes the pose of a person under consideration. [7]

3. Background Idea

Given a depth map, one need not deploy machine learning tool since it's a trivial problem to create a defocused image using depth map. So, we propose an experiment where we feed a completely focused image and a binary depth map so that it can replicate the user action of just marking the object of focus in an image and make the model learn the relative depth map with respect to the object of focus. To help the network reconstruct the output, we extensively use Skip Nets. Before we train the model we need a supervised dataset, in order to help the model learn a segmented depth map or a gradient blurring. The Depth Map we have used is NYU V2 Depth Dataset[8] .The NYU dataset consists of an RGB file and corresponding Depth Map where the gradient of black to white indicates the distance depth from closest to farthest.

3.1. THE NEW DATASET

We have first modified the dataset to Foreground, Midground and the Background of size 640 x 450 as shown in Fig 2. The corresponding binary depth is created by keeping a threshold of 60 on black, which retains all shades of gray and black with below 60 on a scale of 1-256 and converts them to absolute black i.e. 1, where 1 indicates complete

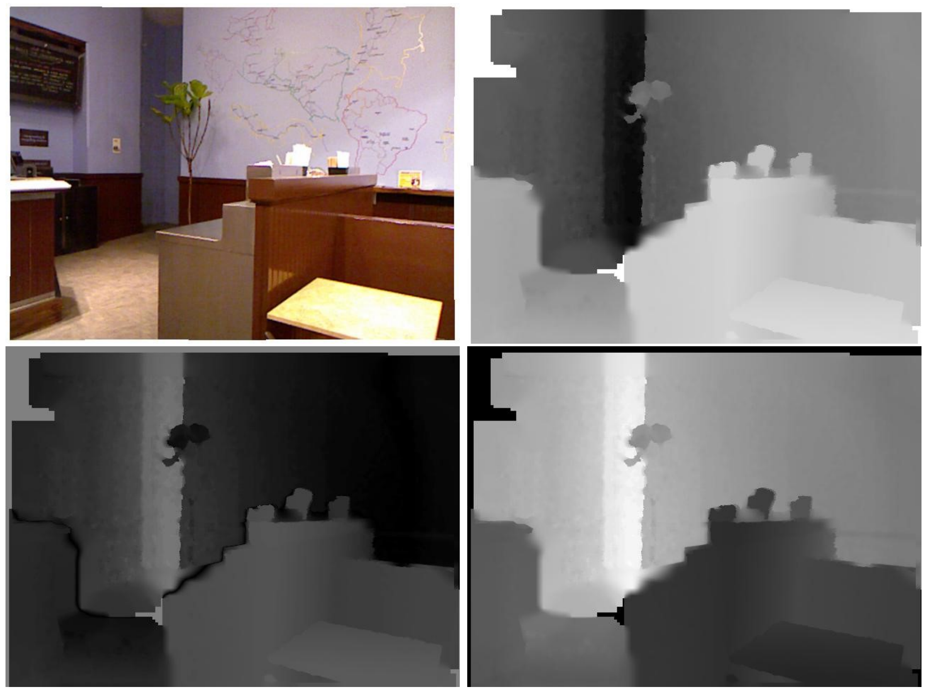

Figure 2. Top-left is the original image from NYU Dataset, top-right is the depth map ”background” created using inversion of original depth map, bottom left indicates the ”mid ground” where gray shade is absolute black and last is the original depth map ”foreground”

black and 255 indicates complete white. Similarly, a patch based dataset is also created, which shall be used in case the model fails to distinguish the focal plane given in an image. Both the sample depth maps have been shown in Fig 3. Along with it, is the original depth map overlaid over the RGB file using Photoshop to create 3 blur maps of the same, where black indicates an object of focus, gray indicates the partial blurring and white indicate complete dense blurring.

The foreground plane depth map is the original depth map provided by the authors of the dataset, where as the ”background” depth map is obtained by inverting the image, $255- > 1$ and $1- > 255$. Whereas the ”Mid plane” depth map is created by considering the mid-range of grey as absolute black and rest colors are computed based on the absolute difference with the mid-tone

Problems with the Dataset

- Unexplained white/black borders provided by the NYU Dataset, hence images are cropped to 600 x 400 pixels
- The Dataset where black patches are around the corners, that will be destroyed when subjected to cropping. The solution involved is to manually inspect all files
- Midplane is mostly black, which implies keeping most of the image in focus. So, such samples are mostly ignored training
- Some of the black patches are unreasonably small, hence can’t be used. Similarly some of the patches after applying the threshold are completely white, hence removed

We rely heavily on the model ability to learn the depth and object segmentation,

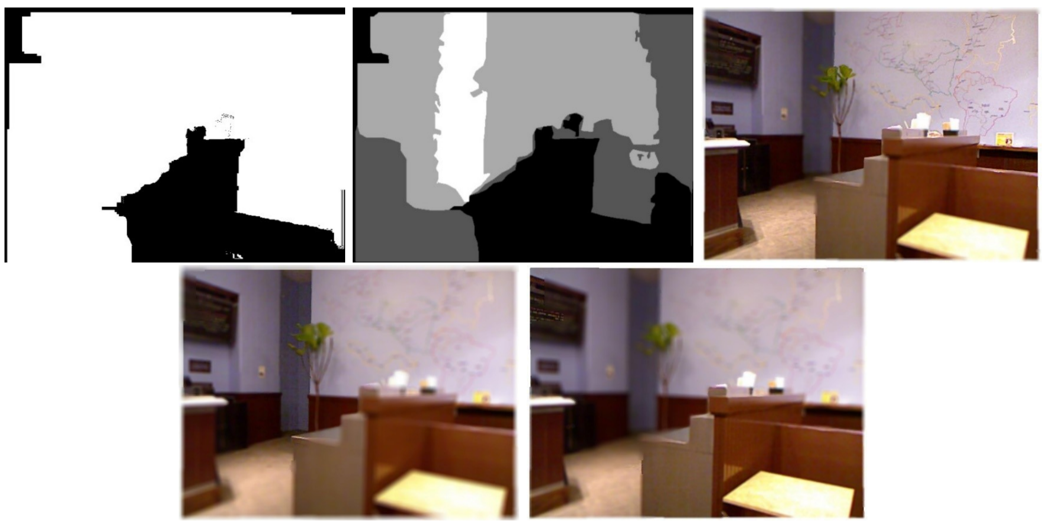

Figure 3. First image is the binary depth map of the RGB image shown in extreme right, and bottom two images. 2nd image is the patch based depth map. The first RGB (extreme right) is the blur map of "mid ground", which is mostly dark, hence most of the image is mostly in focus. The bottom left is the blur map of background where the background is kept in focus and bottom right is the blur map of foreground

in a gradient manner, based on the blurring it has seen in the final image. We expect the final model shall be able to learn the segmentation as well as the relative depth/blurriness based on black patch mentioned. The role of binary depth maps is to replicate the action of the user to mention which part of the image they want to be kept in focus while the model should be intelligent enough to create a smooth defocusing gradient, keeping the segmentation in mind. It's possible that the binary depth map is insufficient for the model to learn the relative depth and defocus based on the object of focus. Hence, we have made a backup of the dataset as a patch based as shown in Fig 3, to give a simple version of depths like black (object of focus), grey simple blurring and white region with complete blurring.

3.2. The Architecture

The Architecture is shown in Fig 4. The figure I is the original Pix2Pix architecture, which is also used as a 4 channel input and rest other are modified versions of it. [9]

4. Execution

The Dataset is mainly created by the MATLAB and Photoshop scripts. We have used a standard Pix2Pix network which uses a convolutional autoencoder, skip nets and 2 encoders sharing a decoder, accompanied by a differentiator.[9]. We have also used a variant of L1 loss function which uses the original depth map for penalization. It penalizes on a scale of 0-2000, where 2000 refers to a black region with max focus. Different gradients of greys are penalized corresponding to their values on a 0-1 scale. Though the model never sees the actual grey depth map it uses it to penalize the reconstruction.

The Model implementations are

- Uniform L1 loss, using two encoders, shown in architecture II

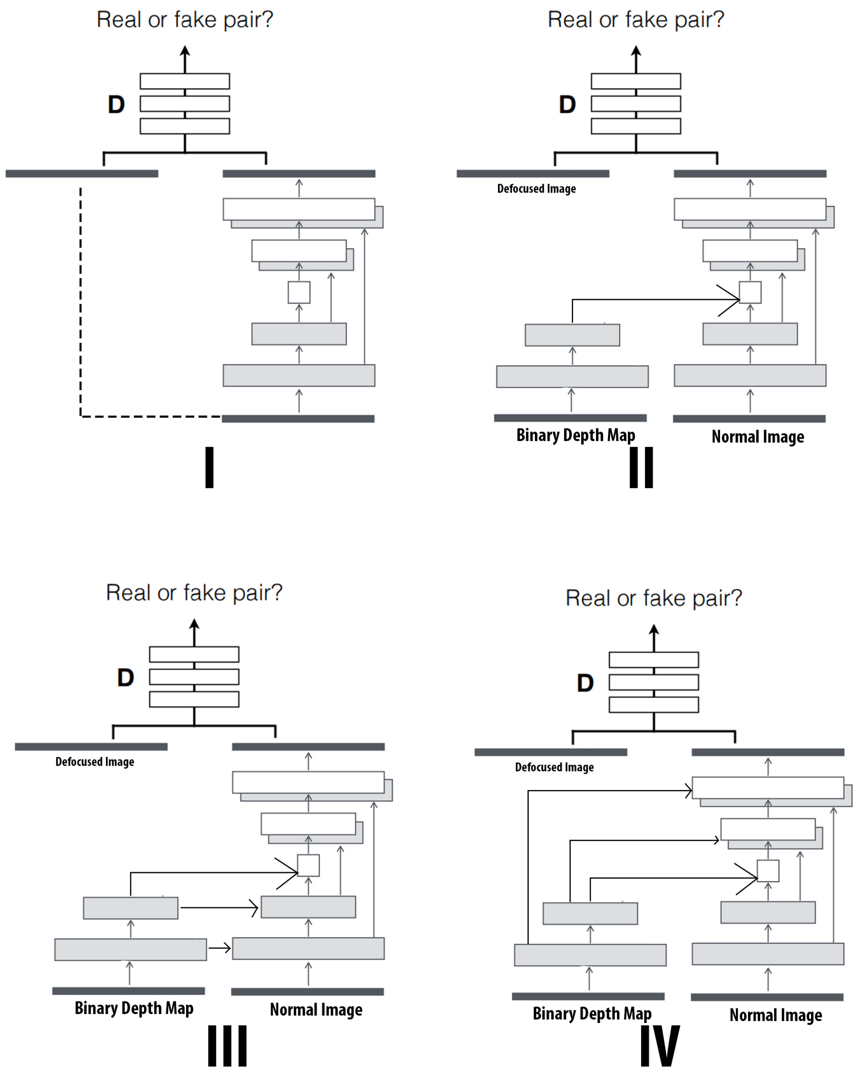

Figure 4. The top left is the original architecture proposed in the Pix2Pix [9], figure II, III and IV are the variants we have applied

- The second implementation uses extra penalization of the black area, the region of the image that need to be kept in focus hence the maximum penalization.
- Third implementation uses the architecture III with the variant of loss as described above but on a patch based depth map.
- The fourth implementation uses the architecture IV with an extra penalty on reconstruction on regions of black. (current on-going)

Since the dataset accepts the fact that it performs well on indoor images, we have 3 test image taken from google[10] and one part of the figure is blackened to inform the model which plane of the image is to be kept in focus. Each test image is similarly converted into 3 depth planes created using photoshop. Fig 5.

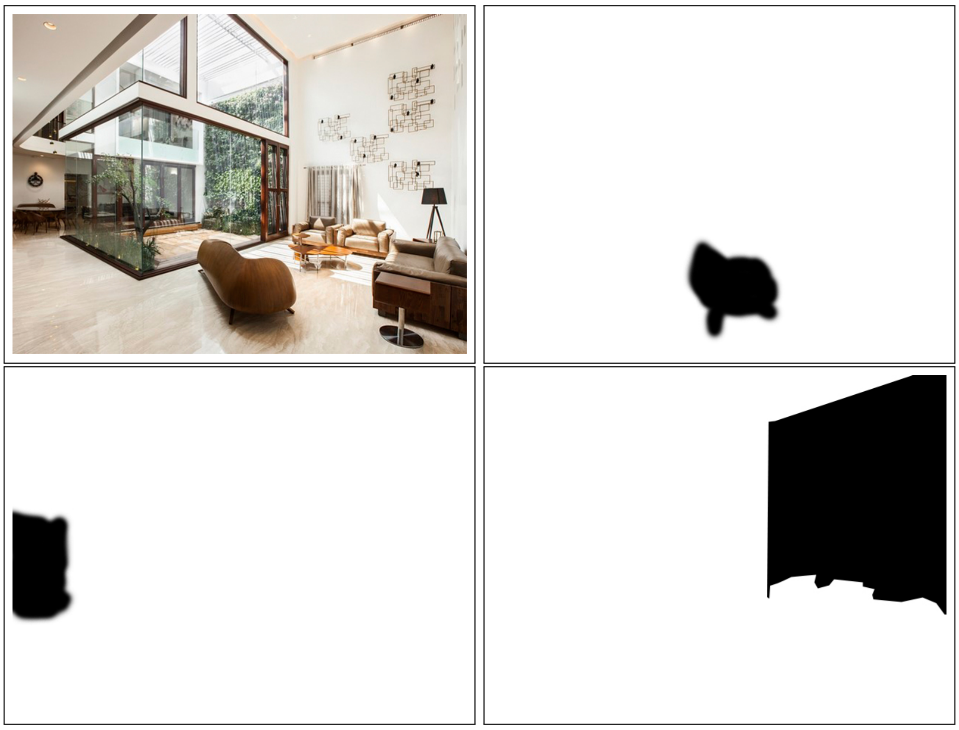

Figure 5. Top left image is from the google [10]. Top right is the focal plane of edge of seat ypical defocused image created from smart phones by overlapping blur map over a RGB photo

5. Results

The model creates a fine gradient blurring on training set but fails on the test set. Results are shown in Fig 6 are for the training set, which shows the network ability to learn the smooth gradient blurring.

Figure 6. Extreme left a random image from training sample from NYU training data set, its corresponding depth map, indicating the tip of chair and the output of Uniform L1 loss, smooth blurring as that of input is achieved

When applied to the test image using the implementation I, one with a uniform loss, actually performs better than rest other implementations. It retains some details while blurring out most of them. Sadly it doesn't differentiate between focal planes and gives almost same output irrespective of depth plane.

The second Model which gives extra emphasis on the black regions of the object of focus performs the worst of all of the implementation. It appears the in this implementation the architecture skips the 2nd encoder learning and gives faulty output, irrespective of all blur planes.

The third implementation is blurry with no focal plane standing out, has some deflection in terms of different focal planes.

Fourth Architecture is not yet completely trained since loss values are fluctuating, rather reaching a stable value.

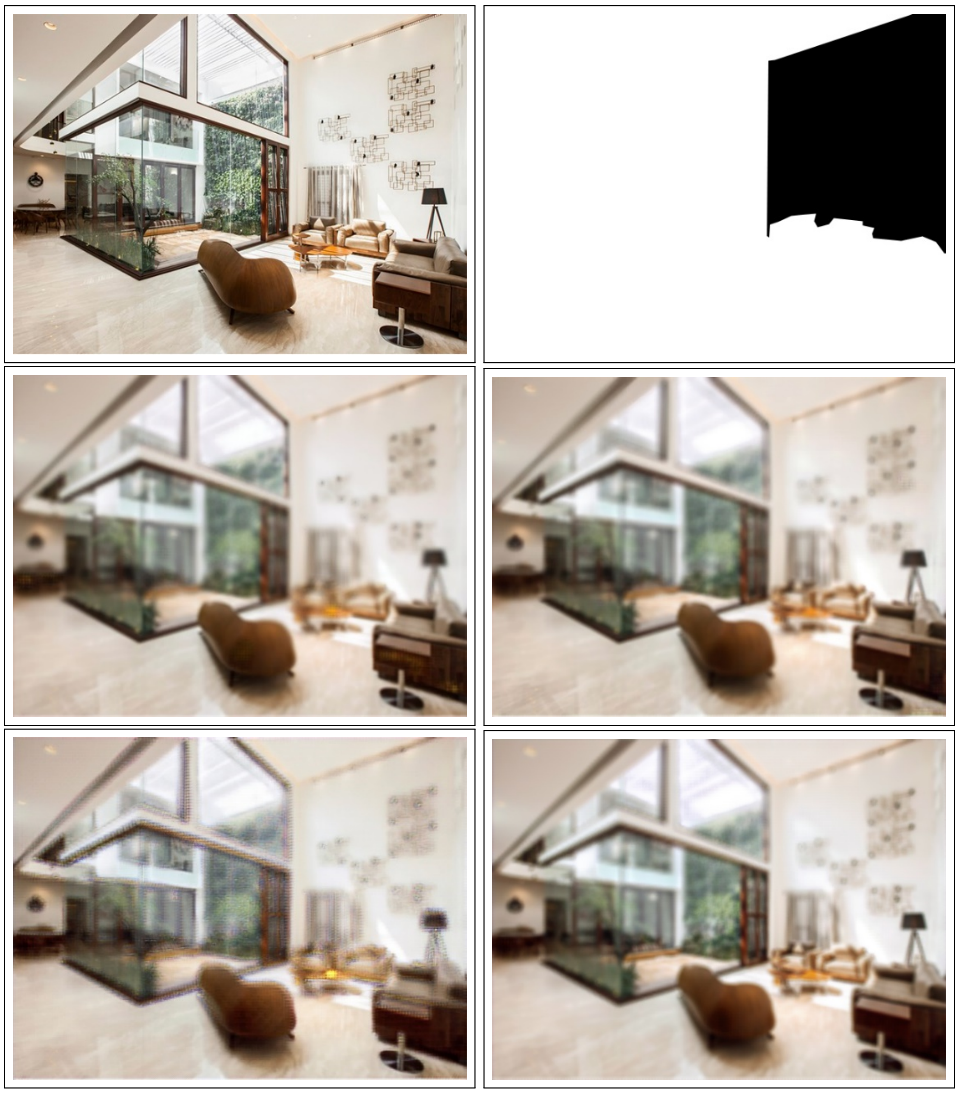

Figure 7. Top left is the input image from google search, Top right is the binary depth created from photoshop, of the fall. Middle left is the output of unifrom L1 loss. Middle right is the output of the L1 loss based on gradient of shades on patch. Bottom left is the output of extra emphaisis of black patch. bottom right is the out put architecture III, with loss function based on the actual depth map.

6. Conclusion and Suggestions

The project is hopeful since it's first of its kind, that can be further developed as an application, where user just need to specify what part they need to keep in focus, which potentially serves a purpose of saving a lot of money on smartphones selling this "portrait mode" as a selling point. The costing is so high in this case as the depth sensor of the secondary lens is way too expensive, thus we believe such a model is able to work, all will be needed is to compress it for a mobile application.

One image is fed using an encoder and the depth map is fed using another encoder, the last approach is to throw away the second encoder and make the image a 4 channel image, use the default Pix2Pix architecture.

The Second implementation is based on the paper [7]. We have a model that can give perfect reconstruction i.e. retain the sharp details. The paper describes an architecture that uses 2 generators, one use binary pose maps to create blurry pose map and the other uses the skip nets to reconstruct the fine details. Potential next step.

7. Future Proposals

- Comparing our created Dataset with the Dataset mentioned in the blog Deep Depth From Focus by Carner Hazirbas, Laura Leal-Taixe, Daniel Cremers [11]
- Use of NYU pretrained model to feed as an input to our interactive plane to produce a smooth depth map based on object of interest specified and create blur by overlapping it with a filter on RGB file

References

[1] The Blurry image, taken from google.
https://tctechcrunch2011.files.wordpress.com/2016/09/berry.jpg

[2] Adobe presentation of their use of AI.
https://www.youtube.com/watch?time$_continue = 20v = DQ8va2ipK8I$

[3] Facebook approach for portrait mode.
https://www.slashgear.com/facebook-thinks-ai-can-replace-expensive-dual-cameraphones-19482748/

[4] Removing blur from a defocused image using deconvolution
http://people.kyb.tuebingen.mpg.de/harmeling/papers/loktyushin2011.pdf

[5] Segmenting focused object in a defocused image
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7420686

[6] Creating fake blur.
http://wscg.zcu.cz/wscg2016/short/A67-full.pdf

[7] Use of binary pose map to create desired pose for human being
https://arxiv.org/pdf/1705.09368.pdf

[8] NYU Dataset
https://cs.nyu.edu/ silberman/datasets/nyu$_depth_v2.html$

[9] pix2pix architecture
https://github.com/yenchenlin/pix2pix-tensorflow
https://arxiv.org/pdf/1611.07004v1.pdf

[10] Test image
https://i.pinimg.com/originals/7c/e2/72/7ce2728ea9d3fdb5c60c1c6d36e72936.png

[11] Test image
http://hazirbas.com/projects/ddff/