# Matlab Out-painting


## ABSTRACT

A classical question in our daily life, where more than 1.2 trillion photos were taken per year, is whether the picture we took were completely meaningful or not ? beside the object that shown in the picture, is the picture background completed and we could see more details about it? the picture is not only capturing memories, but also captures the date and the place from where it was taken, machine learning came along to solve this problem using many algorithms.

As to image out-painting, which is done by fill in the boundaries of picture, our outlook based on deep learning using convolutional neural networks, the training of model through three phases secreted train of DCGAN architecture. For more realistic output, we used local discriminators, that are trained to distinguish real images from completed ones, then the model was able to complete 128*128 images.

The final result is despite the many researches about image inpainting, image out-painting gives more big picture about better processing of pictures in Matlab.



**Keywords**: Deep learning; CNN; GAN; Image processing; Matlab; Image Outpainting; Image Extrapolation.


## 1 INTRODUCTION

Nowadays, we take dozens of images each day, those images can be educative, memorial or leisure, it became sometimes main information form. With the increase of demand, images processing researches try to solve various challenges occur in images, among them image out-painting.

Image out-painting is the filling in boundaries regions of images using information from surrounding areas. We outline in this paper the use of a model for binary in-painting based on the context encoder, also being trained in a completely unsupervised manner, which results fast and efficient out-painting of degraded text, as well as super-resolution of high contrast images. Like auto-encoders, context encoders We propose training Generative Adversarial Networks (GANs)[1], and later reusing parts of the generator and discriminator networks as feature extractors for supervised tasks. GANs provide a path to sophisticated domain-specific data augmentation and a solution to problems that require a generative solution, such image outpainting.

GANs are a smart way to train a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model we train to generate new examples, and the discriminator model which attempts to distinguish examples between the origin and the generated ones. The two models are trained together in an adversarial zero-sum game, the training continues until the discriminator model can't distinguish between the two examples, which means the generator model generates conceivable examples.

We were given in our research source image (m x n), after applying DCGAN algorithm, we would get outpaint image (m x n + 2k where m=128, n=64, and k=32).

Our results were likely naturalistic and convincing output images, panorama too after applying recursive image outpainting.

## 2 RELATED WORKS

The focus in deep learning applications of image processing was centered on recovering and fixing damaged pictures, which was called Image inpaiting, as refer of image completion, However our focus in this paper will be on image outpainting, which is the process of predicting and extrapolation of image in order to to be size-extended with more details and features. The only published paper which is completely related to our work is the paper of Sabini et al.[5] which used deep learning approach to successfully realize outpainting imaging, the model was based on DCGAN architecture, however the source code of the application was in python, while our application coded in Matlab.

The concept of image outpainting isn't much different than image inpainting, so to dive in we should consider the earlier published papers about image completion, which at first based on data-driven perspective as Wang's paper[4], which its operation involves finding suitable content in a candidate image and using it to augment the current image, the output was realistic pictures.

Pathak et al. in the paper "Context encoders: Feature learning by inpainting" elaborated the concept of Context Encoder, a convolution neural network trained to generate the contents of an arbitrary image region conditioned on its surroundings, the paper conclude that a realistic images were successfully generated.

Chintala et al.[6] used the trained discriminators for image classification tasks. However Iizuka et al.[2] used both "local" discriminator, that are trained to distinguish real images from completed ones, and "global" discriminators which looks at the entire image to assess if it is coherent as a whole, bringing on realistic outputs.

Lastly, Both Chintala[6] Liu[3] used convolutional neural network, along with measuring the loss, in order to minimise during image synthesis, which contains two terms for content and style respectively.

## 3 DATASET

For this experiment, we use the images from dataset Places365-Standard [3]. This dataset for image out painting contains 36,500 256×256 images as Figure 1,including various landscapes, buildings, places, rooms and other scenes of everyday life. During the experiment, we employ the same single image for training and testing,while for training ,we have 100 images for validation and train on the remaining 36, 400 images.Besides, We expect our model to be able to overfit on single 128 × 128 color image as opposed to the 512 × 512 image size to speed up training, so we will do some preprocessing in the beginning to downsampled it to 128 × 128 color images.


**Figure 1: small sample of Data set**

### 3.1 PREPROCESSING

In order to prepare our images for training, we use the following preprocessing pipeline. Given a training image $I_{tr}$, we first normalize the images to $I_n \in [0,1]^{128 \text{ x } 128 \text{ x } 3}$.We define a mask $M \in [0,1]^{128 \text{ x } 128}$ such that $M_{ij} = 1 - 1\,[32 \leq j < 96]$ in order to mask out the center portion of the image. Next, we compute the mean pixel intensity $\mu$,over the unmasked region $I_N \odot (1 - M)$.Afterwards,we set the outer pixels of each channel to the average value$\mu$.Formally,we define $I_m = \mu \cdot M + I_m \odot (1 - M)$.In the last step of preprocessing,we concatenate $I_m$ with M to produce $I_p \in [0,1]$ 128 x 128 x 4. Thus,as the result of the preprocessing $I_{tr}$,we output$(I_n, I_p)$.

## 4 METHOD

### 4.1 training pipeline

We had a GAN architecture(G,D) which is based on the paper by Sabini [5].The generator G takes an encoder-decoder CNN, While the discriminator D uses uses strided convolutions to repeatedy downsample an image for binary classification[2].

In each training pipeline as figure 2, We process the each image $I_{tr}$in training dataset to get $I_n$ and $I_P$.Then, we run generator on $I_p$ to get the outpainted image $I_O = G(I_P) \in [0,1]^{128 \text{ x } 128 \text{ x } 3}$.Next, we run the discriminator to classify the image $I_n$ and the outpainted image $I_o$. During the training process,we will calculate the loss and discuss later.

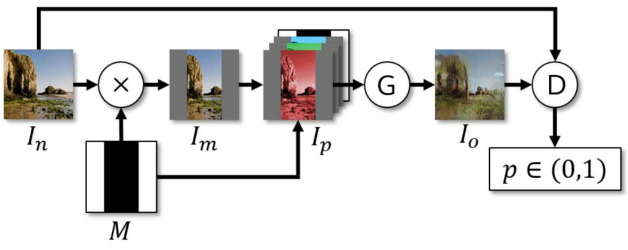


**Figure 2: Training process**

### 4.2 Training schedule

In the original paper[2], they use three-phrase training procedure to facilitate and stabilize the training and define three loss function. In our experiment, we also have three procedures but only show the training loss. The first phase of training(phase 1) conditions the generator by updating the generator weights according to MSE loss for T1 iteration.

The next phase is to update the discriminator weights according to $L_D = -(\log D(I_n) + \log(1 - D(G(I_p))))$ for T2 iterations.

The rest of training(third phase)process for T3 iteration,where the discriminator and generator are trained adversarial according to $L_D$ and $L_G = MSE(I_n, I_p) - \alpha \log D(G(I_p))$. In $L_G$,$\alpha$ is a tunable parameter, which we give it value 0.0004 to control the MSE loss. For other parameter we take Optimize loss for iteration using Adam($lr = 0.001, \beta 1 = 0.9, \beta 2 = 0.999, \varepsilon = 10^{-8}$) according to paper[5]. Our loss function is as following:

$$MSE(I_n, I_p) = \left\| M \odot (G(I_p - I_n)) \right\|_2^2$$

## 5 MODEL

Due to the computational restrictions, we constructed our architecture which is conceptually similar to the paper [2] for outpainting on our dataset. For generator G,we still maintain the encoder-decoder structure from the paper[2] and dilated convolutions to increase the receptive field of neurons and improve the realism.

For the discriminator D, the previous paper [2, 5] use the two kind of discriminators,local discriminator and global discriminator.In our project ,we only use one local discriminator.The discriminator is run on the input image $I_d$ which is equivalent to $I_n$ during training.The output of the discriminator is produced by three parts$D_g(I_d)$, $D_l(I_l)$,$D_l(I_r^{'})$ in concatenator C. $I_l$ is to be the left half of $I_d$,and $I_l^{'}$ is to be the right half of the $I_d$.This can help to ensure that the input to $D_l$ has the outpainted region on the left.Then, the discriminator computes the three part to generate a prediction of the missing part.The final discriminator is as following:

$$P = C(D_g(I_d) \,\|D_l(I_l)\|\, D_l(I_r^{'}))$$

We describe the layers of our model as table 1 and table 2 and table 3.f is the filter size,$\eta$ is the dilation rate, s is the stride,and the n is the number of outputs.Every layer is followed by RELU activation,except for the final output of layer of generator and concatenator which are followed by the sigmoid activation. For the evaluation, although the quality of output image is the best way to evaluate, we still calculate its MSE as our training loss. According to the paper[5] we define it as section 4.2 we have described. For the test, we can add any picture to test the result. Next, we will explain our result.

## 6 RESULT

### 6.1 outpaint single image

Here, we got some outpainted images and list two examples in figure 3. The outpainted desert looks better than the outpainted car.Because the desert image belong to our training dataset ,and the car is randomly chose from another dataset.Therefore, they have different visual effect ion. Even some image during the training

| Type | f | $\eta$ | s | n |
|---|---|---|---|---|
| CONV | 4 | 1 | 1 | 64 |
| CONV | 4 | 1 | 1 | 128 |
| CONV | 4 | 2 | 2 | 256 |
| CONV | 4 | 2 | 2 | 256 |
| CONV | 4 | 1 | 1 | 512 |
| CONV | 4 | 1 | 1 | 512 |
| CONV | 4 | 1 | 1 | 512 |
| CONV | 4 | 1 | 1 | 512 |
| CONV | 4 | 1 | 1 | 512 |
| dilatedCONV | 4 | 1 | 1 | 512 |
| DilatedCONV | 4 | 1 | 1 | 512 |
| CONV | 4 | 1 | 1 | 512 |
| CONV | 4 | 1 | 1 | 512 |
| DECONV | 4 | 2 | 2 | 256 |
| DECONV | 4 | 2 | 2 | 128 |
| CONV | 4 | 1 | 1 | 3 |

**Table 1: Generator G**

| Type | f | s | n |
|---|---|---|---|
| CONV | 5 | 2 | 32 |
| CONV | 5 | 2 | 64 |
| CONV | 5 | 2 | 64 |
| CONV | 5 | 2 | 128 |
| CONV | 5 | 2 | 128 |
| FC | - | - | 1024 |

**Table 2: Discriminator**

| Type | f | s | n |
|---|---|---|---|
| concat | - | - | 2048 |
| FC | - | - | 1 |

**Table 3: concatenation layer**

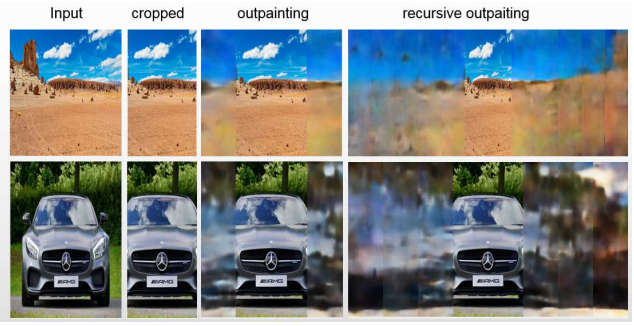


**Figure 3: result of car and desert**

process has the same loss ,the final outpainted images may have different image quality.

In original paper[5],they use the postprocessing to improve the quality of the final outpainted image.Namely,after the renormalization,they blend the unmasked portion using OpenCV's seamless cloning,and output the blended outpainted image. Due to computational complexity, We did not use the postprocessing and get the result not as good as the original paper.

## 6.2 MSE loss

In the figure 4,there is our training loss. We define 25 epochs,this is the result after running 1 epoch, and each epoch has 9465 batches. Because we can not use the computer in the university and our machine is too weak to run, we had to borrow a computer to do the training and create the model, unfortunetly we couldn't save the MSE loss at that time, but later we examine the training for only 1 epoch, so we did not get the enough data of loss and do more validation.But from the loss we can see it decrease rapidly,then almost be stable between 0.04 and 0.14.That means theoretically we got not bad result, while vision still have some improvement.

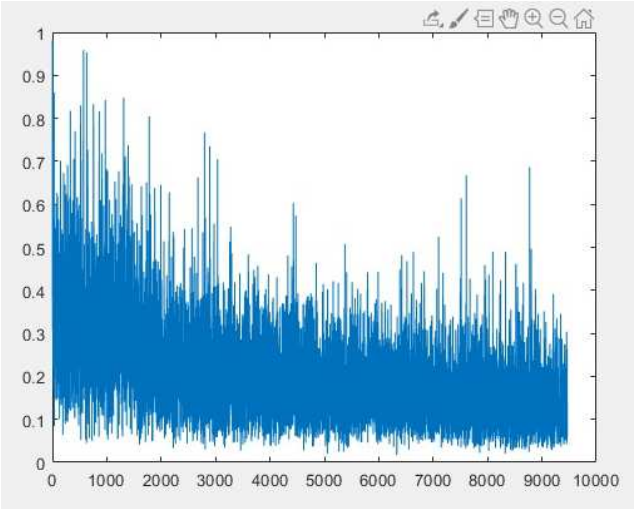


**Figure 4: mseloss**

## 6.3 Recursive outpainting

The outpainted image can be expanded and padding with pixel value to broaden the image.In figure 5,we repeat this process recursively 4 times,then we can get the wider width than input image. Although in this way we can expand the image as we want, the noise tends to repeat with the successive iteration and the border,the edge will become more blurred. Except for this, our model still learning the basic feature and outpaint the sky and the landscape as in the following figure.

## 7 CONCLUSION

In this work, we propose a image outpainting scheme that produces realistic image completion based on Generative adversarial network. The Three-phases we rely on the training were significant

**Figure 5: recursive outpainting 4 times**

for stability, however could be triggered with improvements to become more stable. Our method can outpaint an image to a big-sized extended image and is successfully demonstrated on various scenes.

## 8 FUTURE WORK

Image outpainting can reach improved performance in the future, with some adjustments on the generator algorithm, the architecture could then deliver more realistic images. Further improvements could be by applying adversarial training, also fine-tuning the discriminator's representations, later the algorithm would be able to outpaint the video based on its frames.

## REFERENCES

[1] Pouget-Abadie Jean Mirza Mehdi Xu Bing-Warde-Farley David Ozair Sherjil Courville Aaron C. Goodfellow, Ian J. and Yoshua. Bengio. 2014. Generative adversarial nets. *arXiv*, Article 1804.07723 (July 2014), 36 pages.

[2] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and Locally Consistent Image Completion. *ACM Trans. Graph.* 36, 4, Article Article 107 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073659

[3] Shih K. J. Wang T.-C.-Tao A. Liu G., Reda F. A. and Catanzaro B. 2018. Image inpainting for irregular holes using partial convolutions. *arXiv*, Article 1804.07723 (July 2018), 36 pages.

[4] Y. Liang R. R. Martin M. Wang, Y. Lai and S.-M. Hu. 2014. Biggerpicture: data-driven image extrapolation using graph matching. (July 2014).

[5] Mark Sabini and Gili Rusak. 2018. Painting Outside the Box: Image Outpainting with GANs. arXiv:cs.CV/1808.08483

[6] Alec Radford Luke Metz. Soumith Chintala. 2016. UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS. (July 2016).