International Journal of Computer Trends and Technology (IJCTT) - Volume 68 Issue 5 - May 2020

Outpainting Images and Videos using GANs

Shailendra Singh[1], Nainish Aggarwal[2], Udit Jain[3], Hrithik Jaiswal[4]

[1]*Professor, Department of Computer Science & Engineering, Inderprastha Engineering College, AKTU, INDIA*

[2]*Student(B.Tech), Department of Computer Science & Engineering, Inderprastha Engineering College, AKTU, INDIA*

[3]*Student(B.Tech), Department of Computer Science & Engineering, Inderprastha Engineering College, AKTU, INDIA*

[4]*Student(B.Tech), Department of Computer Science & Engineering, Inderprastha Engineering College, AKTU, INDIA*

[1] shailendra.singh@ipec.org.in
[2] aggarwal.nainish@gmail.com
[3] uditjain350@gmail.com
[4] hrithikjais786@gmail.com

Abstract— *This Outpainting paper studies the main fundamental issue of extrapolation of images or visual context like videos using deep generative models such as GANs (Generative Adversarial Networks), i.e., extending image and video borders with plausible structure and details. In addition, this seemingly simple job faces several critical technical challenges and has its unique properties.*
The challenging task of image and video outpainting (extrapolation) in comparison to it's relative, inpainting (completion), received relatively little attention. So, we followed a deep learning adversarially approach which is based on training a network. The two main problems are the extension of scale and one side constraints.
Extensive studies are carried out on various possible alternati ves and methods connected with them.
We are also exploring our method's potential for various interesting applications that may support work in a variety of fields.

Keywords- *image processing, video processing, generative adversarial networks, extrapolation, outpainting.*

I. INTRODUCTION

Based on limited visual content, humans have the innate ability to perceive invisible environments. For computer vision, this task requires the creation of a form and texture that are semantically meaningful and accurate. In this paper, we concentrate on the particular task of inferring invisible material outside the boundaries of images and videos.

i. Image

For the purpose, outpainting of image also known as image extrapolation, we aim to apply GANs. In this task, we are given an m n source image Is, and we must generate an m n + 2k (In this project, we focus on achieving image outpainting with m = 128, n = 64, and k = 32) image Io such that:

- Is appears in the centre of Io
- Io looks realistic and natural

Image outpainting has been relatively unexplored in literature, but a similar task called image inpainting has been widely studied. Image outpainting is a challenging task, as it requires extrapolation to unknown areas in the image with less neighbouring information. In addition, the output images must appear realistic to the human eye.One common method for achieving this in image inpainting involves using GANs.

Although there are many challenges involved in its implementation, image outpainting has many exciting applications. For example, we can use image outpainting for panorama creation and texture creation.

ii. Video

For the purpose, outpainting of video also known as video extrapolation, we aim to take all the frames of the video and apply GANs. In this task, all the frames we get from video are in similar form of m x n image Is, and we must generate an m x n +2kimage Io such that:

- Is appears in the centre of Io
- Io looks realistic and natural

After extrapolating all the frames, we merge the frames to form a combination of frames which in result give us an extrapolated video. One of the applications of video outpainting is vertically-filmed video expansion.

II. RELATED WORK

GAN. Generative adversarial network (GAN) is a popular approach to generating new data that follow similar distribution as those in the training sets. For many data generation tasks (e.g., face image generation and visual manipulation), GANs have obtained significantly improved results compared to previous approaches, such as variational autoencoders and deep belief networks. Specifically, a GAN contains two networks, a generator G and a discriminator D,

which are trained together (e.g., alternatively). The two networks are adversaries to each other: D aims to tell real inputs from generated ones from G, whereas the goal of G is to maximize the error of D.

The first papers to show image outpainting used a data-driven approach merged with a graph representation of the source image. The researchers were able to achieve realistic results, we anticipate to apply adversarial training for possibly even better outcomes.

A crucial implementation of image inpainting using deep learning by Pathak et al. familiarized the idea of a Context Encoder, a CNN trained adversarially to restructure missing image regions based on nearby pixels. The outcomes presented were comparatively realistic, but still had scope for visual enhancement.

Iizuka et al. enhanced the Context Encoder method for image inpainting by adding an additional discriminator that only took as input the inpainted area and its instant surroundings. This "local" discriminator, combined with the already-present "global" discriminator, permitted the researchers to attain very visually substantial outcomes. As a result, this method is a promising starting point for attaining image outpainting.

Finally, a recent effort in image inpainting by Liu et al. used partial convolutions in combination with the perceptual and style loss first introduced by Gatys et al. Using these methods, the researchers were able to attain extremely accurate outcomes with a fraction of the training required by [4]. In addition, the researchers provided numerous quantifiable metrics that will be useful for assessing our models' performance.

This paper builds upon an initial effort in generative video models. Though, previous work has focused typically on minor patches, and assessed it for video clustering. Here, we develop a generative video model for natural scenes using beaches dataset using state-of-the-art adversarial learning approaches. However, here we are interested in creating short videos with accurate progressive semantics by outpainting each video frame, rather than detecting or retrieving them.

Regarding GAN-based video generation models, Ref. [7] uses an auto-encoding GAN to predict future frames for time-lapse videos. In [4], future frames are predicted using gradient loss to enhance the quality of results. The proposed method is inspired in part by these approaches; nevertheless, predicting future frames is an orthogonal problem.

III. DATASET

In this model, we expect to overfit on a single 128×128 colour images of the beaches. We use a 128×128 image as opposed to the 512×512 image size from to speed up training. For this research, we use the equivalent single image for training and testing.

Our primary dataset for image and video outpainting is composed of 3500 256×256 images from the Beaches dataset . We down-sampled these images and frames of images to 128×128. This dataset is composed of a diverse set of beaches and sceneries as shown in Figure 1.

Figure 1: Beaches Dataset

IV. PROPOSED WORK

A. *Preprocessing of the data*

For the purpose of training our model on beaches dataset, we use a pre-processing pipeline. We use a training image I_t, then we normalize the images to $I_n \in [0, 1]^{128\times128\times3}$ .For the further process we define a mask $M \in \{0, 1\}^{128\times128}$ such that $M_{ij} = 1 - 1[32 \leq j < 96]$ in order to mask out the centre portion of the image.

Now in the next step we take mean pixel intensity μ, over the unmasked region $I_n \odot (1 - M)$. Then, we set the outer pixels of each channel to the average value μ. We define $I_m = \mu \cdot M + I_n \odot (1 - M)$. In the last step of pre-processing, we concatenate I_m with M to produce $I_p \in [0, 1]^{128\times128\times4}$. Thus, as the result of pre-processing I_{tr}, we output (I_n, I_p).

B. *Training Pipeline*

In this paper we have used DCGAN architecture (G, D) similar to that used by Iizuka et al. In this the generator G is used in the form of an encoder-decoder CNN, while the discriminator D uses strided convolutions to repeatedly down sample an image for binary classification.

For each iteration of training, we randomly sample a minibatch of training data. As shown in Figure 2, for each training image I_{tr}, we preprocess I_{tr} to get I_n and I_p, as previously described. We run the generator on I_p to get the outpainted image $I_o = G(I_p) \in [0, 1]^{128\times128\times3}$. Afterwards, we run the discriminator to classify the ground truth (I_n) and

outpainted image (I_o). We compute losses and update parameters according to our training schedule, which will be discussed next.

In video outpainting as shown in Figure 3, we have taken n frames of the video and processed each frame according to Figure 2 and then merged all the frame to form an outpainted video.

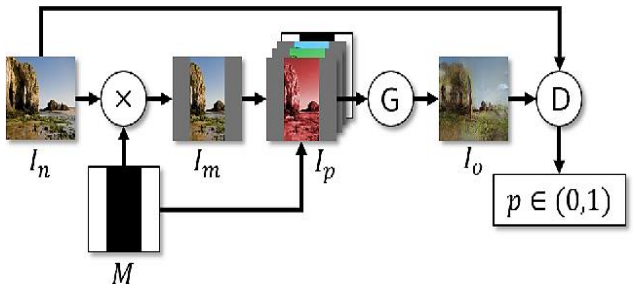

Figure 2: Image Training Pipeline

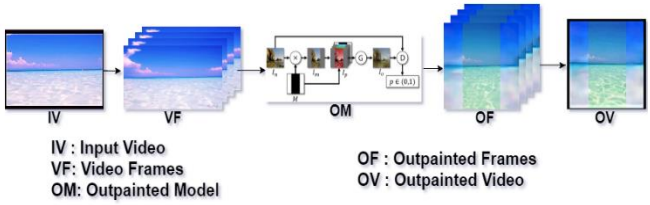

Figure 3: Video Training Pipeline

C. *Training Agenda*

In our training agenda we focused on utilizing three phase training so as to balance the training. Here we have used three different loss functions:

$L_{MSE}(I_n, I_p) = \|M \odot (G(I_p) - I_n)\|$ (1)

$L_D(I_n, I_p) = -[\log D(I_n) + \log(1 - D(G(I_p)))]$ (2)

$L_G(I_n, I_p) = LMSE(I_n, I_p) - \alpha \cdot \log D(G(Ip))$ (3)

In the phase 1 of training, the generator updates the weights of generator in accordance to equation 1 above for T_1 iterations.

In the phase 2 same agenda is followed but with discriminator where it updates discriminator weights in accordance to equation 2 above for T_2 iterations.

Now in last phase all the remaining training is performed for T_3 iterations where both discriminator and generator are adversarially trained.

D. *Our Architecture*

Our architecture proposed is conceptually similar to that of Iizuka et al. [4]. This architecture was proposed for outpainting on Beaches image dataset because of some computational restrictions.

For both discriminator D and generator G we still maintain local discriminator and encoder-decoder structure respectively as in [4] but little modified for image outpainting. Mainly the discriminator is running on an input image I_d (or I_o) during training.

In addition, define I_l to be the left half of I_d, and I_t to be the right half of I_d, flipped along the vertical axis. This helps to ensure that the input to D_l always has the outpainted region on the left. Then, to generate a prediction on I_d, the discriminator computes D_g (I_d), D_l (I_l), and D_l (I_t). These three outputs are then fed into the concatenator C, which produces the final discriminator output $p = C(D_g(I_d) \| D_l(I_l) \| D_l(I_t))$.

Type	f	η	s	n
CONV	5	1	1	64
CONV	3	1	2	128
CONV	3	1	1	256
CONV	3	2	1	256
CONV	3	4	1	256
CONV	3	8	1	256
CONV	3	1	1	256
DECONV	4	1	$\frac{1}{2}$	128
CONV	3	1	1	64
OUT	3	1	1	3

Figure a

(a) Global Discriminator, D_g

Type	f	s	n
CONV	5	2	32
CONV	5	2	64
CONV	5	2	64
CONV	5	2	64
CONV	5	2	64
FC	-	-	512

(b) Local Discriminator, D_ℓ

Type	f	s	n
CONV	5	2	32
CONV	5	2	64
CONV	5	2	64
CONV	5	2	64
FC	-	-	512

(c) Concatenation layer, C

Type	f	s	n
concat	-	-	1536
FC	-	-	1

Figure b

Figure 4: Figure 'a' denotes Generator G & Figure 'b' denotes Discriminator D

E. *Evaluation Metrics*

In evaluation metrics we used Root Mean Square Error (RMSE) as our primary quantitative metrics, though the output of generator is best evaluated.

Given a ground truth image $I_{tr} \in [0, 255]^{128\times128\times3}$ and a normalized generator output image $I`_o = 255 \cdot I_o \in [0, 255]^{128\times128\times3,}$ we define the RMSE as:

$$\text{RMSE}\,(I_{\text{tr}}, I'_{\text{e}}) = \sqrt{\frac{1}{|\text{supp}\,(M)|}\sum_{i,j,k}\quad (M \odot (I_{\text{tr}} - I'_o))^2_{ijk}}$$

F. *Outpainting Flowchart*

A flowchart is a type of diagram that signifies a workflow or process. A flowchart can also be defined as an illustrative representation of an algorithm, a step-by-step method to explaining a task. The flowchart demonstrates the steps as boxes of numerous kinds, and their order by connecting the boxes with arrows.

It is said that a single picture is worth thousands words, the prime advantage of using flowchart is that it is brief as well as simple to understand, another advantage of flowchart is that it presents the information in logical steps which makes it easy for the user to understand the solution logically.

In this paper we have displayed two flowcharts on two different topics image outpainting and video outpainting using same algorithm but using two different approaches.

- Image Outpaint

Our image outpaint flowchart briefs each and every step used in outpainting of an image. Theoretically it is challenging to understand what methods are used and what are their use at particular step, so flowchart helps to understand all the methods used in easier way. Figure 5 shows the image outpainting flowchart.

- Video Outpaint

Similar to the image outpainting flowchart, our video outpainting flowchart also briefs the whole process of usage of each and every method in an easier way. Figure 6 shows the image outpainting flowchart.

Video outpainting uses same algorithm as of image outpainting that is generative adversarial networks but in very less portion as firstly the video is been separated into numerous frames and then the algorithm is applied and at the end the outpainted frames are merged or concatenated with each other in same sequence they were extracted.

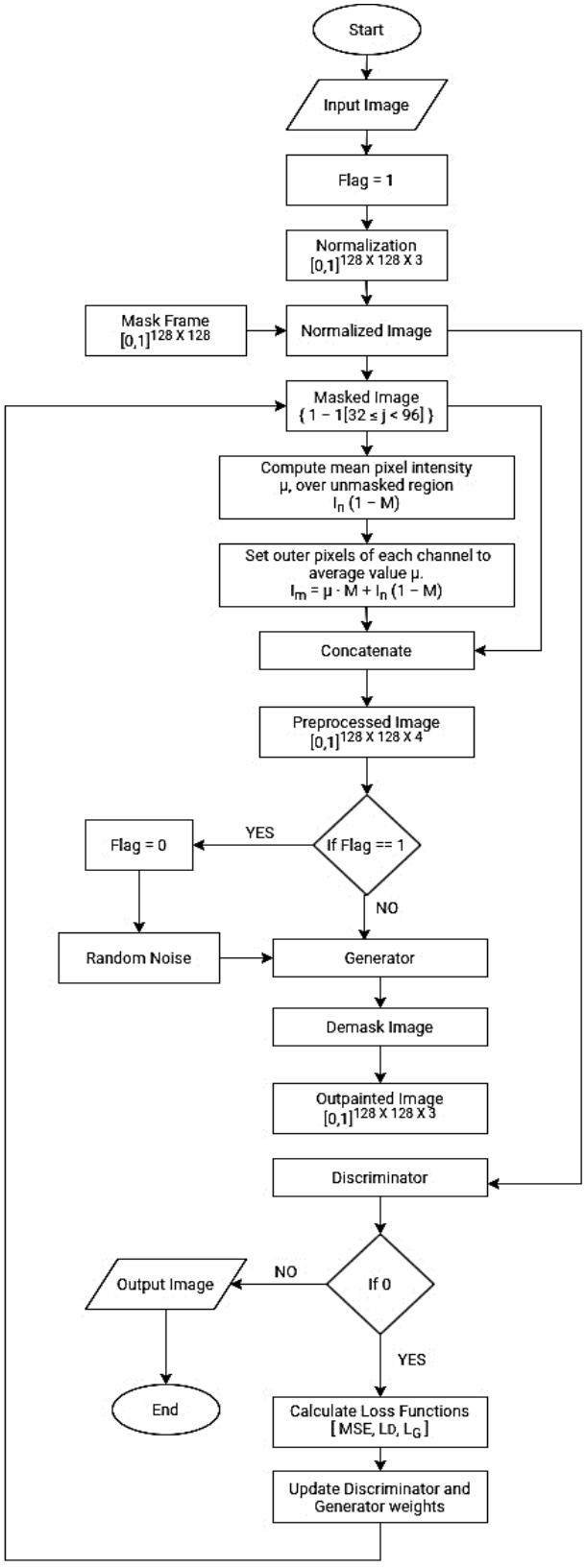

Figure 5: Image Outpainting Flowchart

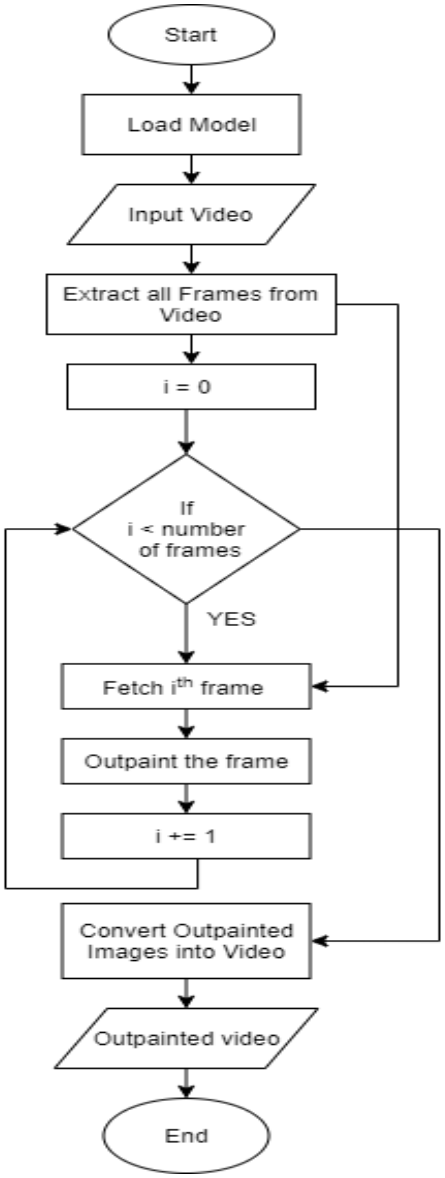

Figure 6: Video Outpainting Flowchart

V. RESULTS

A. *Overfitting on Single Image*

For testing our architecture and training pipeline, we done test of our model on single beach image. Our network was able to overfit to the image, achieving a final RMSE of only 0.885. This shows that our architecture is working fine and can be used for performing image and video outpainting.

B. *Outpainting on our Dataset*

At the end of training on our dataset, we fed images from the validation set through our outpainting pipeline. The final results are shown in Figure 7.

As seen in the third example of Figure 7, the network does not merely copy visual features and lines during outpainting. Rather, it learns to hallucinate new features, as shown by the appearance of a house on the left-hand side.

Figure 7: Outpainting results for a sample of held-out images in the validation set.

C. ***Recursive Outpainting***

Recursive outpainting as the name suggest is the process in which already outpainted image is taken as input. Here we have taken an outpainted image I_o which is sent again as input to the network after expanding and padding with the mean pixel value.

Here we have shown Figure 8 in which this process takes place recursively five times by expanding the image width. As expected, the noise tends to compound with successive iterations. In spite of this, the model effectively learns the overall textures of the image and extrapolates the sky and landscape comparatively realistically.

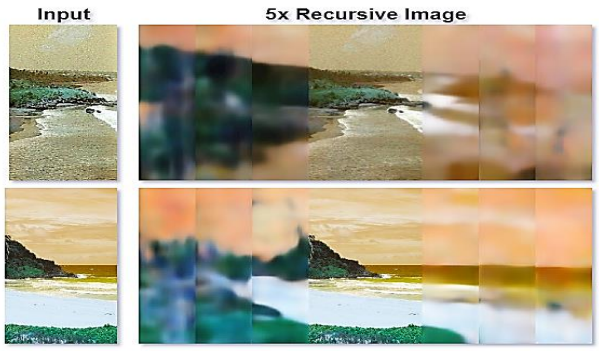

Figure 8: Each right image is the result of recursively outpainting the corresponding left image five times.

VI. CONCLUSIONS

We are able to successfully recognize image and video outpainting applying a deep learning approach. Three-phase

training end up being vigorous during Generative Adversarial Networks (GAN) training. The outcomes from training with only a global discriminator were equally realistic in both the image and video. At the end, we initiate recursive outpainting for image as a means of arbitrarily outspreading an image. Even though image noise compounded with consecutive iterations, the recursively-outpainted image persisted relatively realistic.

VII. FUTURE WORK

Going forward towards future investigation or work, there are numerous potential improvements for our image and video outpainting pipeline. To boost the performance of the model, the generator loss could be augmented with perceptual, style, and total variation losses. To further stabilize training, the Wasserstein GAN algorithm could be incorporated into three-phase training [2].

Further we are investigating to generate 180-degree outpainting from a single image. Also, in continuation to video outpainting, we aim to generate recursive video outpainting by using the same concept used in recursive image outpainting.

VIII. SUPLEMENTARY MATERIAL

The code for our project can be found at https://github.com/Udit9654/Outpainting-Images-and-Videos-using-GANs

REFERENCES

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017. 5

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015. 1, 5

[3] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. ACM Transactions on Graphics (TOG), 36(4):107, 2017. 1, 2, 3

[4] Itseez. Open source computer vision library https://github.com/opencv/opencv , 2015.

[5] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. arXiv preprint arXiv:1804.07723, 2018. 1, 5

[6] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2536–2544, 2016. 1

[7] M. Wang, Y. Lai, Y. Liang, R. R. Martin, and S.-M. Hu. Biggerpicture: data-driven image extrapolation using graph matching. ACM Transactions on Graphics, 33(6), 2014. 1

[8] Saito, Masaki, Eiichi Matsumoto, and Shunta Saito." Temporal generative adversarial nets with singular value clipping." IEEE International Conference on Computer Vision (ICCV). Vol. 2. No. 3. 2017.

[9] Salimans, Tim, et al." Improved techniques for training gans." Advances in Neural Information Processing Systems. 2016.

[10] Suresh Prasad Kannojia, Gaurav Jaiswal "Effects of Varying Resolution on Performance of CNN based Image Classification: An Experimental Study" IJCSE Vol.-6, Issue-9, Sept. 2018.

[11] Sharmila Shaik , Sudhakar P , Shaik Khaja Mohiddin. "A Novel Framework for Image Inpainting". International Journal of Computer Trends and Technology (IJCTT) V14(3):141-147, Aug 2014. ISSN:2231-2803. www.ijcttjournal.org.

[12] Kshitij Tripathi, Rajendra G. Vyas, Anil K. Gupta "Deep Learning through Convolutional Neural Networks for Classification of Image: A Novel Approach Using Hyper Filter". IJCSE Vol.-7, Issue-6, June 2019. https://www.ijcseonline.org/

[13] Charu Khare, Kapil Kumar Nagwanshi "Implementation and Analysis of Image Restoration Techniques". IJCTT - May to June Issue 2011.

[14] Finn, Chelsea, Ian Goodfellow, and Sergey Levine." Unsupervised learning for physical interaction through video prediction." Advances in neural information processing systems. 2016.

[15] S. M. Chavda1, M. M. Goyani "Recent evaluation on Content Based Image Retrieval" IJCSE Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693.

[16] Nazgol Hor, Shervan Fekri-Ershad "Image retrieval approach based on local texture information derived from predefined patterns and spatial domain information" IJCSE. ISSN: 2319-7323 Vol. 8 No.06 Nov-Dec 2019.