CAUSAL DEEP VIDEO INPAINTING

Michael Stecklein
michaelrstecklein@gmail.com

The University of Texas at Austin
Electrical and Computer Engineering
EE 381K Digital Video, Spring 2020
Professor Bovik

ABSTRACT

In this project report, I present a model which performs causal video inpainting using a deep neural network. I create the causal model by applying transfer learning to an existing non-causal model, and evaluate the anticipated degradation in performance both subjectively and qualitatively. I build intuition behind visual discrepancies, and I outline what I have learned from the project. Code and visuals for this project can be found at `https://github.com/mstecklein/DigitalVideoFinalProject`.

1. INTRODUCTION

Deep video inpainting is the process of filling in missing or removed parts of a video in a perceptually plausible way using deep neural networks. It has a variety of applications, such as watermark removal, object removal, occlusion filling, and defect restoration. The field of deep video inpainting is fairly young—all significant papers related to deep video inpainting are less than a year old at this time—and the current state-of-the-art results leave much to be desired. Yet, computational tractability remains a bottleneck when dealing with large neural networks and mass amounts of video. The field of image inpainting has shown exciting results, but the lack of a temporal component makes this task less compute intensive and less complex than that of video inpainting. Since while image inpainted content must be spatially coherent, video inpainted content must also be temporally coherent. Existing models for video inpainting do not explicitly impose temporal coherency, but rather use both future and past frames as inputs to help induce coherency in time. This makes these models non-causal, so even if their inference times were fast enough, they could not be used in real-time applications without significant lag. This report seeks to study causal variations of video inpainting models, for which no future frames are used during the inpainting process. It is expected that this limitation will degrade temporal coherence of the models to some degree. However, the intent of this report is not to fix this degradation, (if it is even possible), but rather analyze its severity and its characteristics.

2. RELATED WORK

2.1. Image Inpainting

Before the rise in popularity of deep neural networks for image processing, the image inpainting task was dominated by patch matching algorithms, which grab textures from other parts of the image to inpaint with. In particular, the randomized PatchMatch algorithm dominated this field of search algorithms [1], but after the advent of GAN-based methods [2, 3, 4], it was quickly outmatched.

The first well-known milestone in deep image inpainting was published in 2017 by Iizuka *et al*. [2]. They used a GAN-like architecture [5] to perform image completion to an unprecedented degree of realism. When refined specifically for faces, their model could generate missing facial features—a task impossible to achieve using patch matching algorithms. Their generator network resembled an encoder-decoder shape, and they used two discriminators. A global discriminator observed the entire inpainted image, while a local discriminator observed only the inpainted region. They attributed a lot of their model's success to their use of two discriminators. Their large network took 2 months and multiple GPUs to train.

In 2018, Yu *et al*. developed a new state-of-the-art GAN model for image inpainting [3]. Their architecture featured a thin and deep network with coarse-to-fine refinement networks, as well as a novel contextual attention layer which encouraged their generator to grab textures from other parts of the image, (much like patch matching). They borrowed the local-global discriminator from Iizuka *et al*. [2], and also outperformed their model, both in realism and model size. Yu *et al*. later further improved their model by proposing gated convolution as a way to generalize the use of masks when inpainting, as well as a new GAN discriminator, SN-PatchGAN, to handle free form masks [4].

2.2. Video Inpainting

The image inpainting model of Yu *et al*. [3] served as a baseline for the first deep video inpainting models, since a novel state-of-art video inpainting model should perform better than the state-of-art image inpainting model, due to the addition of temporal information.

Two initial video inpainting models were developed and published concurrently, one by Kim *et al*. [6] and another by Xu *et al*. [7]. Kim *et al*. [6] published a model which they claimed to be uniquely novel, and which easily exceeded the image inpainting benchmark as well as existing slow optimization-based method of video inpainting. Their model, VINet, was a large LSTM convolutional neural network. It took both past and future frames as an input, and could run at a surprisingly fast 12.5 frames per second. However, their results were not great, especially in hindsight following the publishing of the model of Lee *et al*. [8], which significantly outperformed VINet.

The network of Lee *et al*. [8] used a neural network to compute an alignment transformation for each past and future frame input in order to align these frames with the current frame. (In effect, this is essentially global motion compensation). Then, an encoder-decoder network was used for inpainting. Their novel context matching algorithm sat between the encoder and decoder to combine features based on similarity scores between aligned images. Their model was also implicitly novel in the sense that it did not use optical flow during inpainting. Rather, this allowed their model to pull content from frames further in time to inpaint with, resulting in better performance in applications where motion was slow and/or isolated.

The model by Xu *et al*. [7] was published simultaneously to that of Kim *et al*. [6], but performed subjectively better. At the core of their model was a neural network that inpainted optical flow, as opposed to the RGB pixel values themselves. They titled this network DFC-Net (Deep Flow Completion Network). DFC-Net was composed of 3 smaller subnetworks, called DFC-S, which progressively estimated the optical flow from course to fine, (as originally seen in Yu *et al*. [3]). Each DFC-S was a modified version of a ResNet-50 [9], and took as inputs the optical flows between 5 past and 5 future frames (minus the holes to be inpainted). The "true" optical flows were computed using FlowNet 2.0 [10]. Once the infilled optical flow was computed by DFC-Net, Xu *et al*. [7] used the optical flow field to propagate known pixels outside the fill region inward to inpaint the holes. This iterative procedure was performed both forward in time and backward in time. Finally, any remaining unfilled holes were inpainted by the image inpainting model of Yu *et al*. [3].

It is worth noting that all aforementioned video inpainting models take future frames as inputs, and are therefore not causal. Thus, the study of the effect of forcing causality upon one of these models is motivated.

3. APPROACH

When reviewing the literature of preceding deep video inpainting models, I learned that these models require a lot of compute resources to train from scratch. Many of the previously summarized models took on the order of several days, weeks, or months to train with multiple GPUs. Given the timeline of this semester project and the availability of resources to me as a student, I decided that the best approach for this project would be to apply transfer learning with an existing model to construct my causal model.

Out of the three significant deep video inpainting models, only two can subjectively be called state-of-the-art. Kim *et al*. [6] was outperformed by Lee *et al*. [8], and Lee *et al*. [8] and Xu *et al*. [7] had subjectively similar results. Both of these models can be found publicly. However, Lee *et al*. [8] was not able to publish their training code and loss function due to policies regarding their collaboration with industry. Their loss function and training method was a critical part of their paper and also nontrivial, so performing successful transfer learning with their model would require rebuilding a significant amount of missing code with trial and error. Given the scope of this project, this task seemed too time and compute expensive. Therefore, I decided to use the model by Xu *et al*. [7] as the foundation for my causal model, since all of their source code is publicly available.

The model of Xu *et al*. [7] consisted of a deep flow completion network, DFC-Net, followed by post-processing to inpaint missing regions. DFC-Net is composed of 3 subnetworks, DFC-S, each of which are transfer learned ResNet-50 [9] models. The DFC-S models inpaint missing regions of the flow field from coarse to fine resolution. The post-processing step iteratively inpaints missing pixels using the predicted completed flow field with both forward and backward passes, then fills all remaining missing pixels with the image inpainting model from Yu *et al*. [3].

Figure 1 is copied from the Xu *et al*. [7] and shows a visual summary of their model. I have annotated it to show which pieces of their model would need to change in order to transfer it to a causal model.

To build a causal model, I began by removing the backward propagation and backward merge portions of their post-processing algorithm. This prevented the computed reverse flows from influencing the final pixel values. However, this modification alone was not sufficient to make the model completely causal. Because the DFC-S networks take future optical flow fields as inputs, the completed flow may be more accurate than is possible with only causal inputs, since the network can interpolate between future and past flows. Thus, the second step in making a causal model was to create new subnetworks by removing future flow field inputs from the DFC-S networks. This therefore also required retraining the new networks. The subnetworks were retrained sequentially and individually. The first subnetwork was trained on videos

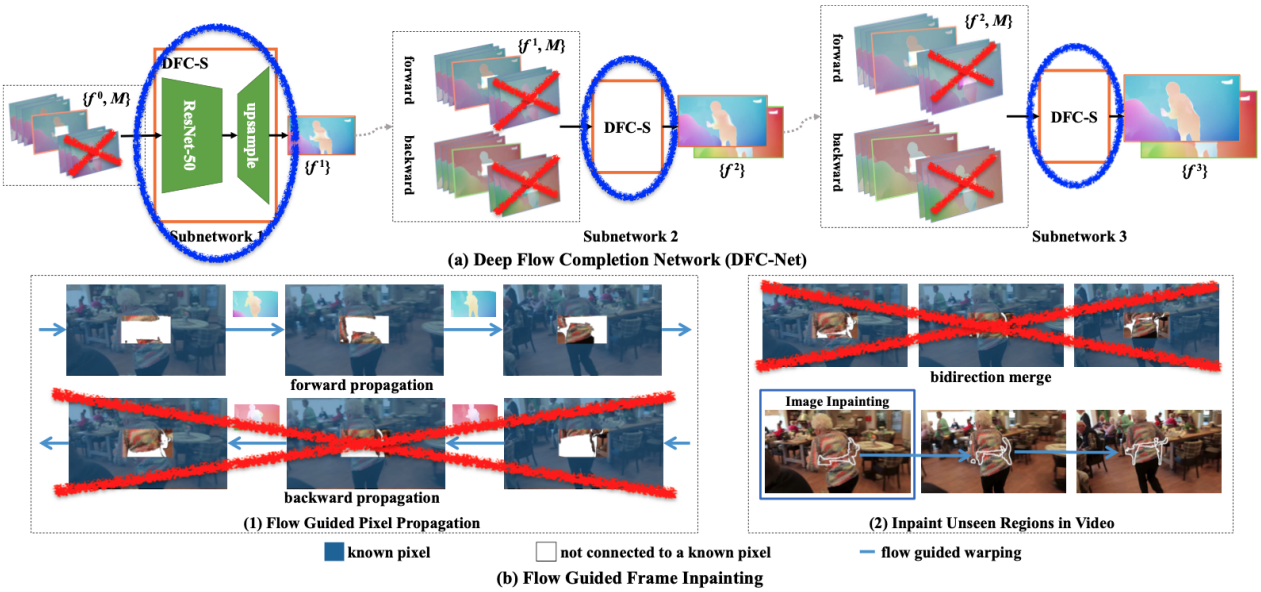

Fig. 1. The pipeline diagram copied from Xu *et al*. [7] is annotated to show changes to their model. The red X's denote pieces removed: non-causal inputs to the DFC-S networks, backward propogation, and bidirection merge. The blue circles highlight networks that were used in transfer learning and were retrained in the causal model.

from the DAVIS dataset [11], and each subsequent subnetwork was trained using the outputs from the previous subnetwork. The subnetworks were trained on the TACC system for around 5 hours each. It was difficult to determine convergence of the networks when training, since the loss was noisy across epochs; the training was stopped when the loss reached a subjectively low stable value.

The resulting pipeline of retrained causal subnetworks and a causal flow inpainting algorithm together constructed a causal deep video inpainting model, built on top of the model from Xu *et al*. [7]. The code and model weights can be found at the link provided in the abstract. The repository is built on top of the v1.1 branch of the repository of Xu *et al*. [7], showing my changes to their original model. The source code of Xu *et al*. [7] can be found at `https://nbei.github.io/video-inpainting.html`.

4. RESULTS

In many video processing fields, such as video inpainting, representative metrics for comparing models do not exist, since the comparisons are based on human perception and are subjective. Therefore, though I will compare PSNR and SSIM scores between models, many of the conclusions I make are based on my own subjective judgements. Also, to understand the full effect of the results discussed in this paper, I would recommend viewing the videos through the GitHub link provided above, since the temporal component cannot be viewed within the format of this paper.

To judge the output of the causal model, I compared three videos from the DAVIS dataset [11]—one of a flamingo walking, one of a lady running, and one of Spiderman in a movie—against the outputs of the Yu *et al*. [3] image inpainting model (frame by frame), the Xu *et al*. [7] non-causal video inpainting model, and my causal video inpainting model. Figure 2 shows the models' results for the three videos. From left to right in each figure, we see the original video, the original video with the masked region highlight (which is to be inpainted), the Yu *et al*. [3] image inpainting model, the Xu *et al*. [7] video inpainting model, and my causal video inpainting model. When played as a video, the image inpainting model shows a significant lack of temporal consistency. The non-causal model by Xu *et al*. [7] inpaints the objects surprisingly well. The causal model performs visually better than the image inpainting model since it maintains temporal consistency. It also performs well on leading edges and small regions. (Take a look at the flamingo's legs and neck. They're mostly gone!). However, its realism is poor particularly towards the center and trailing edges of inpainted regions, (such as the left side of the flamingo and the bottom right of the lady running), creating a "ghosting" effect for removed objects. These are areas where context clues from future frames have the most value. Without them, the model is "guessing" what occluded content might be there in the future, leading to large temporal inconsistencies if it guesses event slightly wrong.

A good way to demonstrate the decrease in context clues in the causal model is to view the videos with inpainting only from the predicted flow fields. In other words, in the post-processing step, any pixels that are not filled in by flow propagation are left black rather than inpainting them with the image inpainting model. This shows us how much influence the predicted flow has on the final output. I show this for

Fig. 2. From top to bottom, we see a sequence of four images for three different DAVIS [11] videos: the first frame of the video, two evenly spaced middle frames, and the last frame. From left to right, we see the original video, the original video with the masked region highlight (which is to be inpainted), the Yu *et al*. [3] image inpainting model, the Xu *et al*. [7] video inpainting model, and my causal video inpainting model.

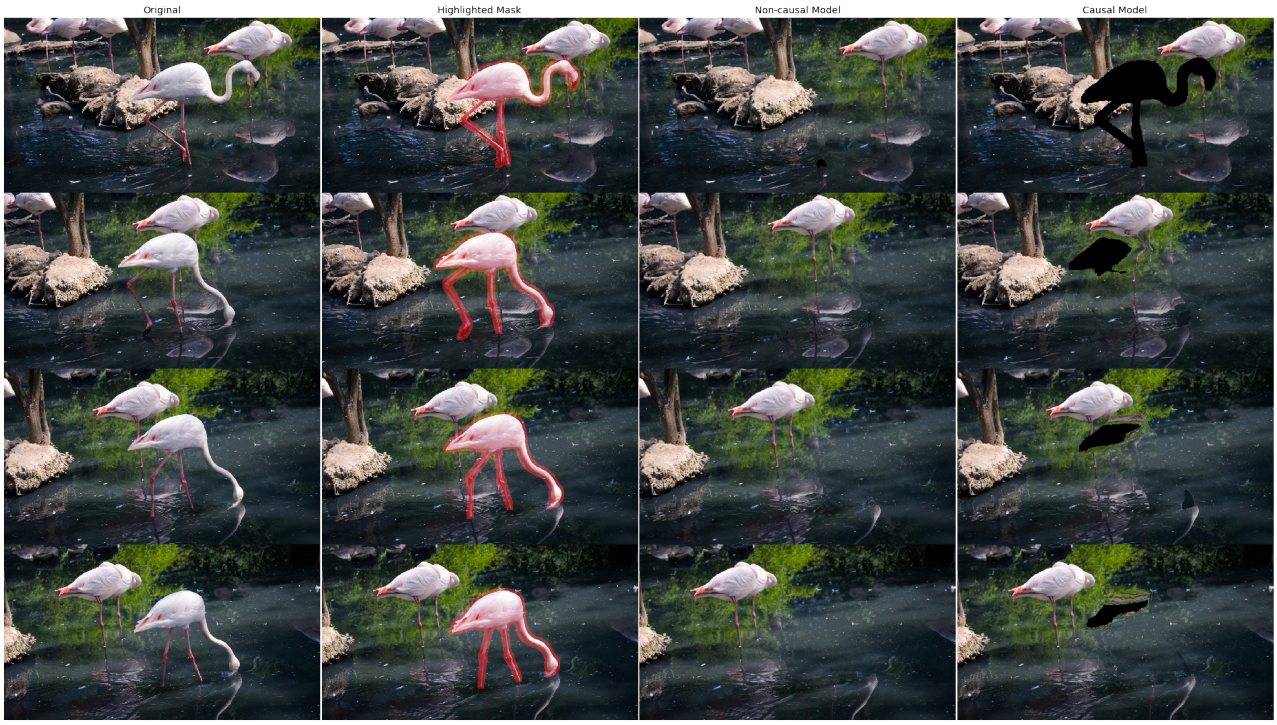

Fig. 3. We see a sequence of four images of the flamingo video inpainted only using flow propagation. From left to right, we see the original video, the original video with the masked region highlight (which is to be inpainted), the Yu *et al*. [3] image inpainting model, the Xu *et al*. [7] video inpainting model, and my causal video inpainting model.

the flamingo video in Figure 3. From left to right, we see the original video, the masked overlay, the non-causal model, and the causal model. It is clear that for this video the non-causal model fills the entire region using pixels propagated by the flow field, resulting in very seamless inpainting. The causal model, on the other hand, leaves a significant portion unfilled. These unfilled regions cause inconsistencies between frames across time. One will notice that for the casual model, the unfilled region is largest at the first frame and decreases as the video goes on. This is due to the fact that the causal model uses previous frames and flow fields to fill in regions, which do not exist at the start of the video. Also, one may observe that within the causally inpainted parts of the frames, the realism decrease as the distance from the leading edge increases. If the flow is less accurate in the causal model, then iteratively propagating pixels will show more inaccuracies the further they propagate. Also, the causal model does not have a reverse pass when propagating pixels, so it cannot "connect" the forward and backward iterations to improve accuracy like the non-causal model does.

If we compare the complete flow fields of the algorithms side by side, we can see the discrepancies that may cause unrealistic pixel propagation. From center to right, Figure 4 shows the "true" flow computed by FlowNet 2.0 [10], the completed flow by the non-causal model, and the completed flow by the causal model. When viewed across time, it is clear that the optical flow from the causal model is noisier than the true flow and the non-causal model. The centers of inpainted regions flicker with light colors, representing fast changing motion vectors of large amplitude. This therefore explains the lack of realism of pixel propagation far from inpainted regions' boundaries.

Up to this point, all results and conclusions have been visually subjective and based on intuitions. Now, we compare the models' outputs quantitatively, but with metrics that are not ideal. Our goal with inpainting is to achieve perceptual realism to a human, which is difficult to quantify. Previous papers have no good suggestions for metrics, and simply use PSNR and SSIM themselves. But since PSNR is a measure of difference between the original pixel values and the inpainted pixel values, it is not necessarily representative of our goal, since it has little correlation to realism. (And when removing objects from a scene, the point is to make the pixels different anyways!). Likewise, since SSIM measures structural similarity, it is not necessarily representative either. Nevertheless, Table 1 shows a comparison of the PSNR and SSIM values for the image inpainting model, the non-causal model, and the causal model. These values were averaged across all frames in the flamingo, lady running, and Spiderman test videos. The non-causal model is superior in both metrics. Interestingly, the causal model performs worse than just image inpainting in PSNR but better in SSIM. This implies that the causal model didn't fill colors as accurately but holds shapes better. Since these metrics are computed on a frame by frame basis, they also do not measure temporal consistency; if they did, I would imagine that the image inpainting model would perform far worse.

Lastly, I observed that the causal model performed partic-

Fig. 4. From top to bottom, we see a sequence of four images for three different DAVIS [11] videos: the first frame of the video, two evenly spaced middle frames, and the last frame. From center to right, we see the "true" flow computed by FlowNet 2.0 [10], the completed flow by the non-causal model, and the completed flow by the causal model.

Model	PSNR	SSIM
Frame Inpainting, Yu *et al*. [3]	23.083	0.924
Non-causal, Xu *et al*. [7]	23.417	0.936
Causal, mine	22.543	0.928

Table 1. Qualitatively comparison of models' performance: baseline frame-by-frame inpainting, the non-causal model by Xu *et al*. [7], and mine.

ularly poor for the first few frames of each video, then improved over time. This is because the causal model derives all its features from previous frames, which do not exist at the start of a video. To visualize this, Figure 5 shows the the SSIM values for the three videos over time for both the causal and non-causal models. Though it is subtle, the SSIM scores of the causal model tend to get closer to the scores of the non-causal as time increases. (This is more noticeable in the running lady and Spiderman plots).

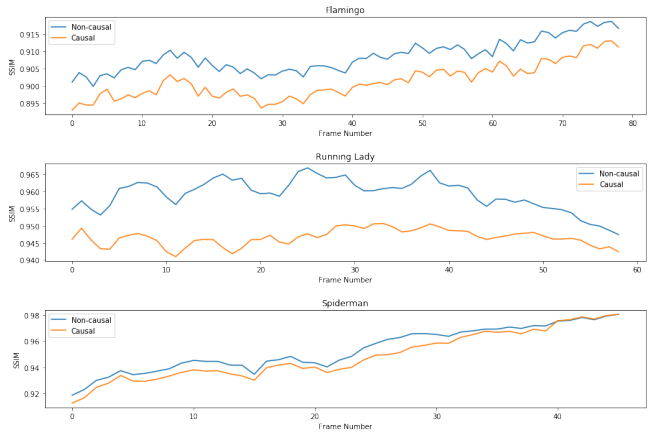

Fig. 5. Comparison of SSIM scores across time between the non-causal model of Xu *et al*. [7] and my causal model for the three test videos.

5. WHAT I LEARNED

Foremost, this project was my first experience working on a non-trivial video processing task. In that sense, it was an educational exercise in practical video processing. I learned that video processing can be quite tedious at times, since it often takes a long time to implement small changes when working with a lot of data. However, the final results are often very rewarding as well, since there is a unique satisfaction in viewing the results of a video processing model you've built yourself.

This project was also my first experience working with a sizable neural network. Most notably, I learned that there is an "art" to working with NNs. Often, the network architectures that produce the best results are achieved through trial and error, possibly without strong intuition as to why one worked better than the others. A GPU and plenty of time is a MUST for working with NNs. They take way more compute power and time to train than I imagined, and it is not feasible to train large NNs on a CPU. More practically, I learned how to use popular libraries for neural networks, GPUs, and computer vision, such as PyTorch, TorchVision, mmcv, Nvidia's CUDA, and so forth.

Throughout this project I read several papers outlining different deep CNN models, and in doing so I noticed a trend across successful models. The models which performed best were those that were built upon some intuition, rather than just brute force. Models which just connected an input to a desired output with a multitude of dense layers generally did not perform well compared to models that used NNs for specific, intuitive steps in a larger algorithm. Similarly, models which were careful to limit unintended restrictions on their networks saw benefits. For example, in [4], Yu *et al*. improved their own model from their first paper [3] by removing an accidental restriction that carried a boolean mask deep into the network's layers. This observation was interesting to me, because the mentality around neural networks often seems to be one of "plug-and-chug", i.e. to blindly give a network inputs and outputs and let it figure out the mapping. However, this was not my observation throughout this project.

Lastly, and most importantly, I learned the importance of considering humans' visual perception in video processing applications. When looking to improve a model, especially when no relevant metrics exists, brain-storming ideas from the perspective of human perception is the only perspective that isn't blind to the target. In this project specifically, perceptual intuitions for optical flow and temporal consistency helped me identify weaknesses in my causal model and contrast it with its non-causal parent. The target audience of videos is always humans, so it is humans that they must be designed for.

6. CONCLUSION

Overall, the results of the causal model were poorer than the non-causal model, (as expected), but more temporally consistent than image inpainting frame by frame. (AGAIN, I would strongly encourage the reader to view the results as a video/GIF at: `https://github.com/mstecklein/DigitalVideoFinalProject`). Intuitive explanations were given for all inaccuracies seen in the causal model, most of which stem from less accurate optical flow fields. Given more time and computation, further improvement may be made by confirming convergence of training in all three subnetworks, as well as attempting to extend other non-causal models. Through this project, I developed practical skills and built intuitions for both video processing and neural networks.

7. REFERENCES

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.

[2] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa, "Globally and Locally Consistent Image Completion," *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, vol. 36, no. 4, pp. 107, 2017.

[3] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang, "Generative image inpainting with contextual attention," *CoRR*, vol. abs/1801.07892, 2018.

[4] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang, "Free-form image inpainting with gated convolution," *CoRR*, vol. abs/1806.03589, 2018.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," 2014.

[6] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon, "Deep video inpainting," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy, "Deep flow-guided video inpainting," *CoRR*, vol. abs/1905.02884, 2019.

[8] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim, "Copy-and-paste networks for deep video inpainting," 2019.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[10] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," *CoRR*, vol. abs/1612.01925, 2016.

[11] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 724–732.