

RT-GAN: Recurrent Temporal GAN for Adding Lightweight Temporal Consistency to Frame-Based Domain Translation Approaches

Shawn Mathew^{1*}, Saad Nadeem^{2*}, and Arie Kaufman¹

¹ Stony Brook University, New York, USA

² Memorial Sloan Kettering Cancer Center, New York, USA

*Equal Contribution. Corresponding Author: nadeems@mskcc.org

Abstract. Fourteen million colonoscopies are performed annually just in the U.S. However, the videos from these colonoscopies are not saved due to storage constraints (each video from a high-definition colonoscopy camera can be in tens of gigabytes). Instead, a few relevant individual frames are saved for documentation/reporting purposes and these are the frames on which most current colonoscopy AI models are trained on. While developing new unsupervised domain translation methods for colonoscopy (e.g. to translate between real optical and virtual/CT colonoscopy), it is thus typical to start with approaches that initially work for individual frames without temporal consistency. Once an individual-frame model has been finalized, additional contiguous frames are added with a modified deep learning architecture to train a new model from scratch for temporal consistency. This transition to temporally-consistent deep learning models, however, requires significantly more computational and memory resources for training. In this paper, we present a lightweight solution with a tunable temporal parameter, RT-GAN (Recurrent Temporal GAN), for adding temporal consistency to individual frame-based approaches that reduces training requirements by a factor of 5. We demonstrate the effectiveness of our approach on two challenging use cases in colonoscopy: haustral fold segmentation (indicative of missed surface) and realistic colonoscopy simulator video generation. We also release a first-of-its kind temporal dataset for colonoscopy for the above use cases. The datasets, accompanying code, and pretrained models will be made available on our Computational Endoscopy Platform GitHub (<https://github.com/nadeemlab/CEP>).

Keywords: Temporal GAN · Colonoscopy · Domain Translation

1 Introduction

More than 14 million colonoscopies are performed every year, just in the U.S. Even though there is a LIVE video feed guiding the navigation of the endoscopist during the 20–40 min minimally-invasive procedure, hardly any of these videos are stored for later analysis due to storage constraints/costs (each

high-definition video can be several gigabytes); instead a few relevant frames are stored for reporting/documentation purposes only. To help assist endoscopists during procedures for tumor detection or to document the quality of the procedure for education purposes (e.g. by tracking the colon surface area missed during procedure via haustral fold occlusion – higher surface area missed equates to higher possibility of missed cancer), AI models are normally trained on the few stored frames easily accessible via electronic health records (through Institutional Review Board approval). *Rather than trying to train video models from scratch with vast amounts of video data (NOT available), can we add lightweight temporal consistency to our best-performing single-frame models for video analysis to aid endoscopists?* We address this question in this paper.

Recently, unsupervised domain translation methods have shown promising results across different colonoscopy tasks (e.g. to translate between optical [OC] and prior-treatment virtual/CT colonoscopy [VC]), but not all have been extended to video. The domain translation models that have been extended to video create new models from scratch to accommodate video sequences. Once a frame-based model has been finalized, one can either try simple post-processing normalization across frames to get “quasi-consistency” [11] or train a new model from scratch with full temporal consistency. The first approach is only possible on very specific tasks, such as depth estimation, where there is one correct result. Tasks such as realistic image generation cannot be concatenated together with simple approaches (results will flicker as shown in **supplementary video**). The second, more general option however requires significantly more computational and memory resources for training. Moreover, temporally-consistent unsupervised video-to-video domain translation (RecycleGAN [1] derivatives) typically requires learning both directions of translation when only a single direction may be relevant, for example, colonoscopy to depth, colonoscopy to fold segmentation, synthetic to real colonoscopy simulation, etc. This forward and backward learning with temporal components increases the number of learnable parameters by several orders of magnitude. Even still, the general approaches like RecycleGAN may not utilize domain specific knowledge that can vastly improve results. Incorporating domain specific contributions from frame based models can be quite involved and time consuming.

In this work, we present Recurrent Temporal Generative-Adversarial Network (RT-GAN) for adding lightweight temporal-consistency to unsupervised image-to-image domain translation models (that reduces training requirements by a factor of 5). RT-GAN allows traversal between temporal consistency and fidelity to the frame-based models using a single tunable weight parameter while focusing on a single translation direction. Specifically, RT-GAN uses recurrent information by referencing the previous frame and its result as seen in Figure 1c. A temporal discriminator takes the generator’s results for 3 consecutive frames to build temporal consistency; using only 3 sequential frames for temporal learning was a design choice to minimize resource utilization. In essence, RT-GAN builds on the representations learned by any unsupervised image-to-image domain translation model and adds temporal consistency to these with-

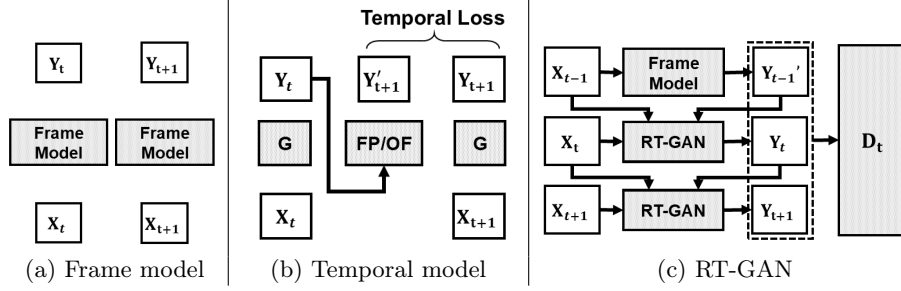


Fig. 1. Depicting how temporal consistency can be added. X is the input video and Y is the resulting video output. (a) Frame-based model, (b) Temporal consistency is added in RecycleGAN [1] and OfGAN[17] using optical flow or future frame prediction. (c) RT-GAN uses three consecutive output frames from generators are passed into a discriminator to provide temporal consistency. The first frame, Y'_{t-1} is generated from a fully trained frame-based model. The other two frames are created by RT-GAN to be temporally consistent with Y'_{t-1} .

out needing to redesign task-specific components. We demonstrate the effectiveness of RT-GAN in adding lightweight temporal consistency to two frame-based models, FoldIt [9] haustoral fold segmentation model with some inherent “quasi-consistency” across frames and CLTS-GAN [10] color-lighting-texture-specular reflection augmentation model with no consistency at all across frames.

Related Work Bashkirova et al. performed a number of experiments using a CycleGAN model that uses 3D convolutions with varying input types such as randomly sorted frames, ordered frames, and frames stacked as a 3D tensor. They found that using the stacking frames into 3D tensors provided the best results at the cost of extra training requirements [2]. Bansal et al. proposed RecycleGAN [1], a network for unsupervised video retargeting, that does not require any task-specific modules and adds temporal consistency components (such as optical flow) on CycleGAN to extend it to videos (Figure 1b). Specifically, an additional future frame prediction network is added for temporal consistency. This increases memory requirements especially since two predictor networks are needed, one for each domain. OfGAN [17] predicts optical flow using an architecture similar to the one shown in Figure 1b to translate synthetic colonoscopy sequences to real colonoscopy video sequences; OfGAN relies on texture, lighting, and specular reflection information to be embedded in the input videos to generate realistic colored output sequences. CycleSTTN [3] is another recent work that learns a video domain translation task for specular augmentation, however, it requires paired data generation which may not be feasible for all tasks.

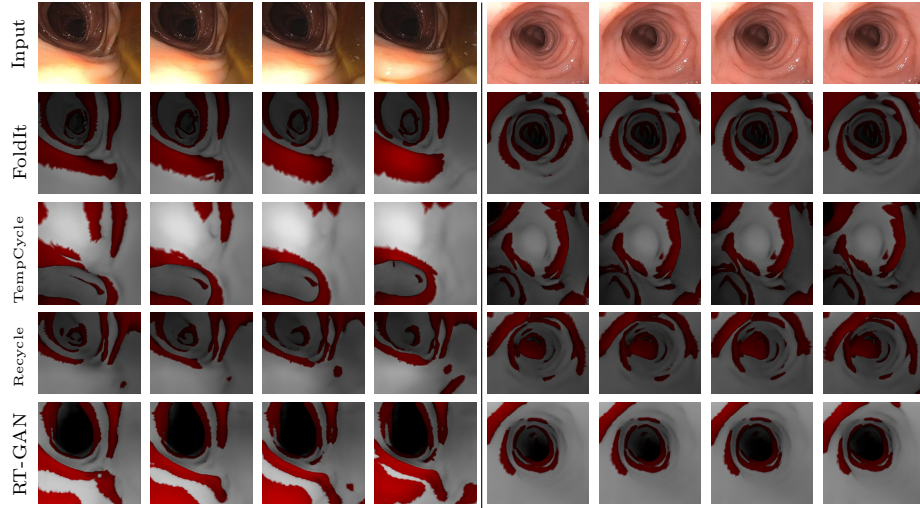


Fig. 2. Comparisons of the results for RT-GAN (Ours) with stitched images from FoldIt, TempCycleGAN, and RecycleGAN on optical colonoscopy video dataset from [7]. Full results are found in the **supplementary video**.

2 RT-GAN: Recurrent Temporal GAN

Dataset: The OC and VC dataset was created from 10 patients at Stony Brook University Hospital that had VC procedures followed by OC procedures. The OC videos were cropped to a size of 256x256 to remove borders in the frames created by the fish-eye lens in the colonoscope. The videos for VC were created from triangulated meshes of the colon extracted from CT scans as described by Nadeem et al. [12]. A virtual camera flies through the mesh with random rotations and lights at both sides of the camera. To better replicate the conditions of the colonoscopy procedure, the inverse square fall-off property is applied to the lights [8]. The videos for both the VC and OC datasets were split into 300 sets of 3 sequential frames. In total, training, validation and testing datasets are composed of 1500, 900 and 600 frame triplets respectively. Hausrat fold segmentation data is generated in a similar manner to Mathew et al. [9]. The VC 3D meshes will be publicly released as well for video generation via Blender or VR-CAPS[6].

Methods: Typically for unsupervised domain translation, at least 2 generator networks are being updated during training time. One generator learns the translation between the input domain and output domain, while the other learns the inverse direction. Typically, only one generator is required to provide the domain translation results for an application. RT-GAN only trains one generator reducing resources during training (see Table 1). RT-GAN builds off of the results from a fully trained frame-based model, F . The results of the frame-based model can

be pre-computed, so it does not affect the required resources for training. Resource requirements will be defined by the more resource hungry frame-based models. The RT-GAN’s generator G , translates from the input domain X to the output domain Y . G takes 3 images as input to produce the output y'_t . The first input is the frame, x_t that is to be translated. The next input is the previous frame in the input sequence, x_{t-1} , to give the network context and a better understanding of motion. The last input image for G is y'_{t-1} , the result for x_{t-1} . y'_{t-1} gives the generator context on the previous frame with which the output needs to be temporally consistent with. The input for RT-GAN’s generator can be seen in Figure 1c.

G is trained using two discriminators, each having its own adversarial loss. The adversarial/discriminator loss is described below:

$$\mathcal{L}_{adv}(G, D, y, y') = \log(D(y)) + \log(1 - D(y')), \quad (1)$$

where y' is from the generator and y is from the training data.

The first discriminator, D_t , learns temporal consistency. D_t compares a 3 frame sequence from the output domain to a 3 frame sequence created from the generators. The first frame in the triplet is provided by F , while the next 2 temporally consistent frames are provided by G . G aligns its results with F in order to provide temporal consistency, but F ’s results is independent of G . The temporal adversarial loss is described as,

$$\begin{aligned} \mathcal{L}_t(G, F, D_t, Y, X) = \\ \mathcal{L}_{adv}(G, D_t, \{y_{t-1}, y_t, y_{t+1}\}, \{F(x_{t-1}), y'_t, y'_{t+1}\}), \end{aligned} \quad (2)$$

where y'_t is $G(x_{t-1}, x_t, F(x_t))$ and y'_{t+1} is $G(x_t, x_{t+1}, y'_t)$.

A separate discriminator, D_f , ensures that G ’s results appear similar to F . It compares the paired input and output frames for F and G . The adversarial loss for D_f is described as:

$$\mathcal{L}_f(G, F, D_f, X) = \mathcal{L}_{adv}(G, D_f, \{x_t, F(x_t)\}, \{x_t, y'_t\}) \quad (3)$$

If the camera and the subject do not move between two frames, the resulting output of the model should also should not change. A stationary loss \mathcal{L}_s is added for this which also helps enforce the model to use the previous frame’s output (y'_t) rather than just predicting from the current frame. The stationary loss is defined as:

$$\mathcal{L}_s(G, X) = \|y'_t - G(x_t, x_t, y'_t)\|_1, \quad (4)$$

where $\|\cdot\|$ represents the ℓ_1 norm. *Note that the stationary loss differs quite substantially from a perceptual loss. The stationary loss ensures temporal stability, while a perceptual loss is meant to improve the quality of the image.*

The complete objective function for the network is:

$$\mathcal{L}_{obj} = \lambda \mathcal{L}_t(G, F, D_t, Y, X) + \mathcal{L}_f(G, F, D_f, X) + \mathcal{L}_s(G, X) \quad (5)$$

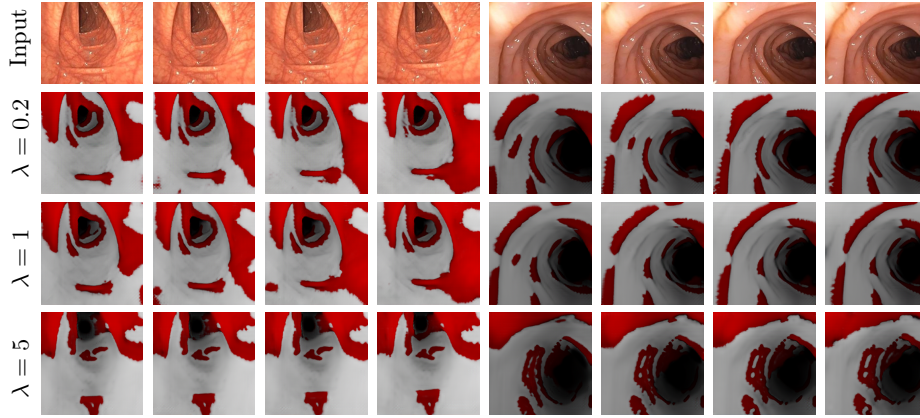


Fig. 3. Results for varying temporal weights (λ). The first row is the input and the second row shows $\lambda = 0.2$. As λ decreases RT-GAN’s is more faithful to FoldIt. The next row shows $\lambda = 1$ where there is a balance between temporal and the frame losses. The last row shows $\lambda = 5$. Here the annotation shapes tend to remain consistent between frames. Full videos are found in the **supplementary video**.

Table 1. Number of learnable parameters (in millions) and training time per epoch for RecycleGAN [1], TempCycleGAN[4], OfGAN [17], FoldIt [9], CLTS-GAN [10], and RT-GAN (ours). Models were trained on NVIDIA Quadro RTX 6000 GPU.

	Learnable Parameters	Training Time
RecycleGAN [1]	137.11	~ 742 s
TempCycleGAN [4]	46.65	~ 836 s
OfGAN [17]	142.23	~ 947 s
FoldIt [9]	82.14	~ 700 s
CLTS-GAN [10]	55.15	~ 857 s
RT-GAN (Ours)	25.22	~ 394 s

where λ is a tunable weight to determine the tradeoff between the temporal smoothness and fidelity to the frame-based model. D_t is a PatchGAN discriminator with 3D convolutions to help it learn temporal information while D_f is a PatchGAN discriminator with 2D convolutions to learn spatial information. G uses a Resnet architecture with 9 blocks. *The bottleneck for training parameters is determined by the frame-based model. Trainable parameters is the $\max(\text{frame_model}, \text{RT-GAN})$ and training time is $\text{frame_model} + \text{RT-GAN}$.*

3 Results and Discussion

The training time and memory usage of RT-GAN is analyzed in Table 1. RT-GAN reduces the number of learnable parameters by a factor of 5 while decreasing the training time by half when compared with RecycleGAN and OfGAN.

Table 2. Quantitative results on a synthetic colon dataset [9] with two textures and ground truth fold annotations. The consistency column indicates the frame-based geometric consistency of the model despite different textures as described by Mathew et al.[11]. These sequences are shown in the **supplementary video**.

	Text 1 (IoU/DICE)	Text 2 (IoU/DICE)	Consistency (IoU/DICE)
RecycleGAN	0.34/0.21	0.33/0.20	0.76/0.63
FoldIt	0.47/0.31	0.50/0.33	0.77/0.64
RT-GAN	0.55/0.39	0.54/0.38	0.81/0.69

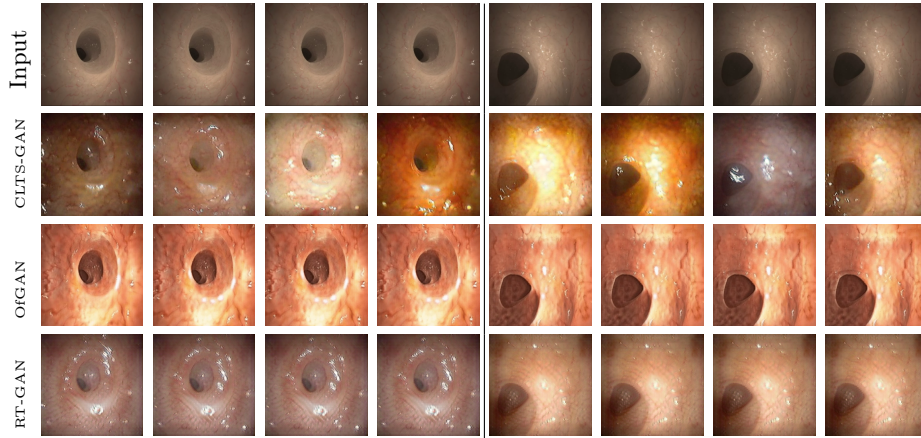


Fig. 4. Results for RT-GAN trained on CLTS-GAN. The top portion shows results on rendered mesh frames. CLTS-GAN’s results change drastically over time. RT-GAN builds off CLTS-GAN to provide consistent specular and texture between frames. The bottom half shows results using OfGAN’s input video, which embeds texture and specular information. CLTS-GAN adds more intricate specular reflections and textures, and RT-GAN inherits this property. OfGAN relies on the embedded texture and specular to produce its output. Full videos are in the **supplementary video**.

Compared to TempCycleGAN, a video domain translation model with a minimal amount of image generators, RT-GAN reduces the number of learnable parameters and training time by a factor of 2. FoldIt, a frame-based model for haustral fold segmentation, uses fewer resources than RecycleGAN as it deals with individual frames. RT-GAN still requires lesser resources than FoldIt because it only learns one direction of translation while FoldIt learns four [9]. CLTS-GAN [10] only learns two directions of translation, so RT-GAN reduces the learnable parameters in half. When training RT-GAN, the hardware requirements are capped by the frame-based model since RT-GAN requires lesser resources.

To test the effectiveness of RT-GAN in fold segmentation context (indicative of the total missed surface during colonoscopy), we added RT-GAN on top of FoldIt haustral fold frame-based model [9]. In Figure 2, we compare RT-GAN, FoldIt, TempCycleGAN, and RecycleGAN results on public video sequences from

Ma et al. [7]. RecycleGAN has many variants, however sifting through all the variants and applying task-specific components requires great effort on part of the end users. We chose RecycleGAN for comparisons since it has all the base temporal components seen in the more advanced variants and is not task-specific. As shown in supplementary video, FoldIt and RecycleGAN both had jittery results and FoldIt occasionally smooths out the deeper parts of the endolumen. In contrast, RecycleGAN translated these deeper endolumen parts as folds since it does not contain any task-specific modules or losses. RT-GAN utilizes the task-specific modules from FoldIt while providing temporal consistency. TempCycleGAN is more consistent, however, similar to RecycleGAN it doesn't have the task specific additions and it fails to accommodate the deeper portions of the endoluminal view. Complete videos sequences are shown in the **supplement**.

For quantitative analysis, synthetic colon dataset with ground truth annotations was used [9]. Table 2 shows that RT-GAN's additional temporal consistency provided improvement on the IoU and DICE scores for both textures. RT-GAN is also more consistent than the other models despite different textures. Additionally, the optical flow can be compared in the input sequences and output sequences as done by Rivoir et al. [14]. The mean difference between the input optical flow and output optical flow on our textured colons for RecycleGAN, FoldIt, and RT-GAN are 2.4788, 0.9021, and 0.8479, respectively. This indicates that RT-GAN can better capture the motion between frames when compared with other models like RecycleGAN and FoldIt. The synthetic colon results can be found in the supplementary video. In Figure 3, the λ parameter to control temporal consistency is shown. When λ is set to a lower value, it tries to be more faithful to FoldIt. As λ is increased, RT-GAN makes the annotations smoother so it looks more temporally consistent.

We also evaluated RT-GAN on real colonoscopy video generation/simulation using the frame-based CLTS-GAN model [10]. CLTS-GAN creates colonoscopy frames with different colors, lighting, textures, and specular reflections using noise parameters. For real colonoscopy video generation/simulation, RT-GAN was trained for 200 epochs on 1800 frame triplets of colonoscopy video and 3D renderings of the colon using virtual colonoscopy from [10]. The results of real colonoscopy video generation from synthetic sequences are shown in Figure 4. The top half shows video generation from virtual colonoscopy renderings. CLTS-GAN's use of noise parameters allows it to generate drastically different output across frames. RT-GAN is much smoother and the specular reflections and textures are consistent; in the **supplementary video**, the overall color and lighting changes over time since RT-GAN only looks at the previous frame (and doesn't have a longer-term memory, an issue we will resolve in the future). The bottom portion of Figure 4 compares (RT-GAN + CLTS-GAN) with OfGAN [17]. OfGAN is confined to creating textures and specular reflections that are embedded in its input video. In contrast, CLTS-GAN adds additional texture and specular reflections but lacks temporal consistency. RT-GAN uses CLTS-GAN's texture and specular information and adds temporal consistency on top of it. Complete video results are shown in the **supplement**.

Limitations. The first is the lack of long term memory since RT-GAN only receives information from the previous frame. Incorporating transformers or State Space Models [5, 13, 16, 15] could mitigate this issue. Additionally, RT-GAN can inherit some of the limitations of its frame based model, e.g. FoldIt cannot handle frame occlusion and hence both FoldIt and RT-GAN can hallucinate endoluminal view for occluded frames. In the future, we will explore the application of RT-GAN to other endoscopy procedures, such as cystoscopy, bronchoscopy, and naseopharyngoscopy.

Acknowledgments. This project was supported by NIH R37CA295658, MSK Cancer Center Support Grant/Core Grant (P30 CA008748), and NSF grants CNS1650499, OAC1919752, and ICER1940302.

Disclosure of Interests. Dr. Nadeem has received speaker honorarium from Roche Tissue Diagnostics which is not related to this work. He also serves on MONAI (Medical Open Network for Artificial Intelligence) consortium advisory board, unrelated to this work. Other authors do not declare any competing interests.

References

1. Bansal, A., Ma, S., Ramanan, D., Sheikh, Y.: Recycle-gan: Unsupervised video retargeting. *Proceedings of the European Conference on Computer Vision (ECCV)* pp. 119–135 (2018)
2. Bashkirova, D., Usman, B., Saenko, K.: Unsupervised video-to-video translation. *arXiv preprint arXiv:1806.03698* (2018)
3. Daher, R., Barbed, O.L., Murillo, A.C., Vasconcelos, F., Stoyanov, D.: Cyclesttn: A learning-based temporal model for specular augmentation in endoscopy. *International Conference on Medical Image Computing and Computer-Assisted Intervention* pp. 570–580 (2023)
4. Engelhardt, S., De Simone, R., Full, P.M., Karck, M., Wolf, I.: Improving surgical training phantoms by hyperrealism: deep unpaired image-to-image translation from real surgeries. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* pp. 747–755 (2018)
5. Fuoli, D., Huang, Z., Paudel, D.P., Van Gool, L., Timofte, R.: An efficient recurrent adversarial framework for unsupervised real-time video enhancement. *International Journal of Computer Vision* pp. 1–18 (2023)
6. İncetan, K., Celik, I.O., Obeid, A., Gokceler, G.I., Ozyoruk, K.B., Almalioglu, Y., Chen, R.J., Mahmood, F., Gilbert, H., Durr, N.J., Turana, M.: VR-Caps: A virtual environment for capsule endoscopy. *Medical Image Analysis* p. 101990 (2021)
7. Ma, R., Wang, R., Pizer, S., Rosenman, J., McGill, S.K., Frahm, J.M.: Real-time 3D reconstruction of colonoscopic surfaces for determining missing regions. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* pp. 573–582 (2019)
8. Mahmood, F., Chen, R., Durr, N.J.: Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *IEEE Transactions on Medical Imaging* **37**(12), 2572–2581 (2018)
9. Mathew, S., Nadeem, S., Kaufman, A.: Foldit: Haustral folds detection and segmentation in colonoscopy videos. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* pp. 221–230 (2021)

10. Mathew, S., Nadeem, S., Kaufman, A.: CLTS-GAN: color-lighting-texture-specular reflection augmentation for colonoscopy. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* pp. 519–529 (2022)
11. Mathew, S., Nadeem, S., Kumari, S., Kaufman, A.: Augmenting colonoscopy using extended and directional CycleGAN for lossy image translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 4696–4705 (2020)
12. Nadeem, S., Kaufman, A.: Computer-aided detection of polyps in optical colonoscopy images. *SPIE Medical Imaging* **9785**, 978525 (2016)
13. Neimark, D., Bar, O., Zohar, M., Asselmann, D.: Video transformer network. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* pp. 3163–3172 (2021)
14. Rivoir, D., Pfeiffer, M., Docea, R., Kolbinger, F., Riediger, C., Weitz, J., Speidel, S.: Long-term temporally consistent unpaired video translation from simulated surgical 3D data. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* pp. 3343–3353 (2021)
15. Ryali, C., Hu, Y.T., Bolya, D., Wei, C., Fan, H., Huang, P.Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., et al.: Hiera: A hierarchical vision transformer without the bells-and-whistles. *Proceedings of the International Conference on Machine Learning (ICML)* (2023)
16. Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., Baik, S.W.: Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE Access* **6**, 1155–1166 (2017)
17. Xu, J., Anwar, S., Barnes, N., Grimpen, F., Salvado, O., Anderson, S., Armin, M.A.: Ofgan: Realistic rendition of synthetic colonoscopy videos. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* pp. 732–741 (2020)