Realtime Background Replacement and Super Resolution for Video Conferencing Applications

Adityan Jothi, Aditi Hoskere Deepak, Srujana Subramanya

EE599 : Deep Learning - Fall 2020

Mentor: Prof. Brandon Franzke

Contents

Introduction

Figure 1: Original Image

Figure 2: Low resolution background replaced image

Figure 3: High resolution on background replaced image

Courtesy: https://heartbeat.fritz.ai/how-to-replace-backgrounds-in-photos-with-machine-learning-6396fc518384

Outline

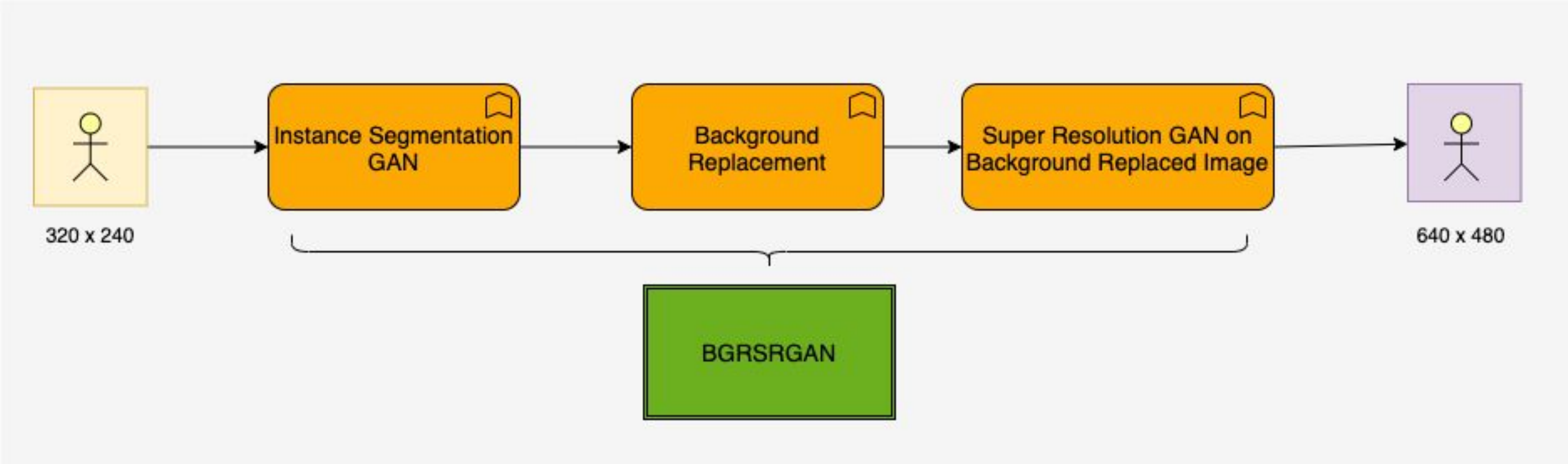
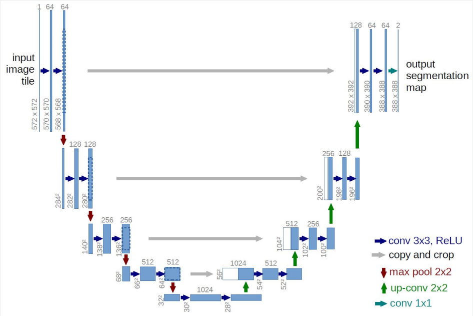


Figure 4: Illustration of project overview.

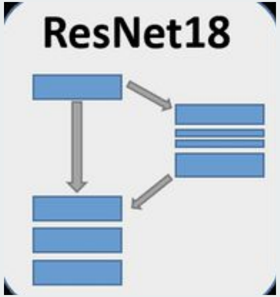
Instance Segmentation for Background Replacement

Input



UNet Generator Network

+



Discriminator Network (Pretrained - resnet18)

Output

Figure 5: UNet GAN Architecture.

Image courtesy of Ronneberger et al.

What we did?

- Dataset used: COCO Dataset - 2014.
- Initial design inspired by UNET-GAN
- Uses MSE Loss for Generator and Wasserstein Loss for Discriminator
- Changes made:
    - Used Resnet18 as discriminator
    - Trimap generation
    - Foreground extraction

Trainable Paramaters	Forward/Backward Pass (Mb)	Batch Size	Training Time on p3.2xlarge (mins/epoch)
2,89,57,481	1200.33	10	12

Table 1: Model Summary

Results

Figure 6: Input Image

Figure 7: Trimap

Figure 8: Intermediate Results of our model

Figure 9: Final Results of our model

Results

Figure 10: Input Image

Figure 11: Trimap

Figure 12: Intermediate Results of our model

Figure 13: Final Results of our model

Results

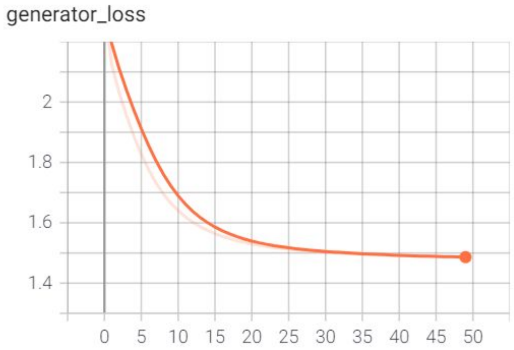


Figure 14: Generator Loss

Figure 15: Discriminator Loss

Figure 16: Total Loss

Super Resolution

Do DNNs hallucinate in high resolution?

- Process of recovering High Resolution (HR) image from a Low Resolution (LR) image.

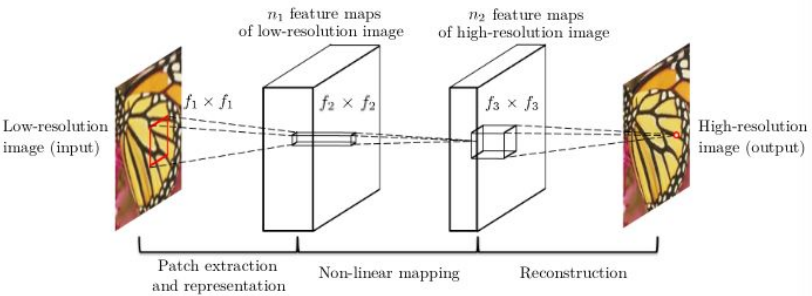


Figure 17: Overview of Super Resolution.

Courtesy: “Image Super-Resolution Using Deep Convolutional Networks”, C.Dong, C. C. Loy, K. He, et al. IEEE Transactions on Pattern Analysis and Machine Intelligence 2016

What we did?

- Dataset:
  - Youtube videos from game streamers, podcasters (Video call-esque nature of data)
  - 9500+ samples from 1080p video used at 240p -> 480p super resolution factor for training with larger batch size
  - Dataloader loads high resolution images X_hr, we do lr_transform(X), hr_transform(X) to generate the inputs and targets of the network respectively.

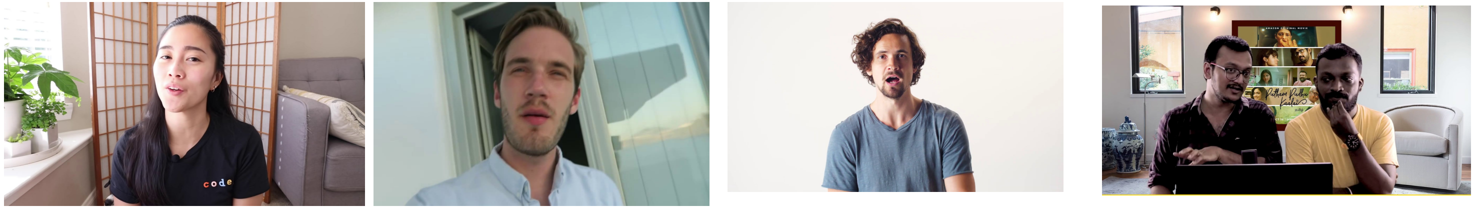


Figure 18: Some samples from the SR Dataset that was created Low Resolution @(320x240) High/Super Resolution @(640x480).

What we did?

- Initial design inspired by EDSRGAN and Fast-SRGAN
- Uses Perceptual Loss and Adversarial Loss for Generator
- Changes made:
  - Depthwise Convolutions to reduce parameter size for decreasing inference time
  - Swish activation instead of ReLU for better performance
  - Used Resnet18 as discriminator
  - Used Resnet50 as feature extractor for computing perceptual/content loss

Trainable Parameters	Forward/Backward pass size (Mb)	Batch Size	Training time on g4dn.4xlarge (mins/epoch)	Best PSNR (dB)
467,843	2144.53	8 (pretrained), 4 (trained)	26.25	61.3

Table 2: Model Summary

Model Architecture

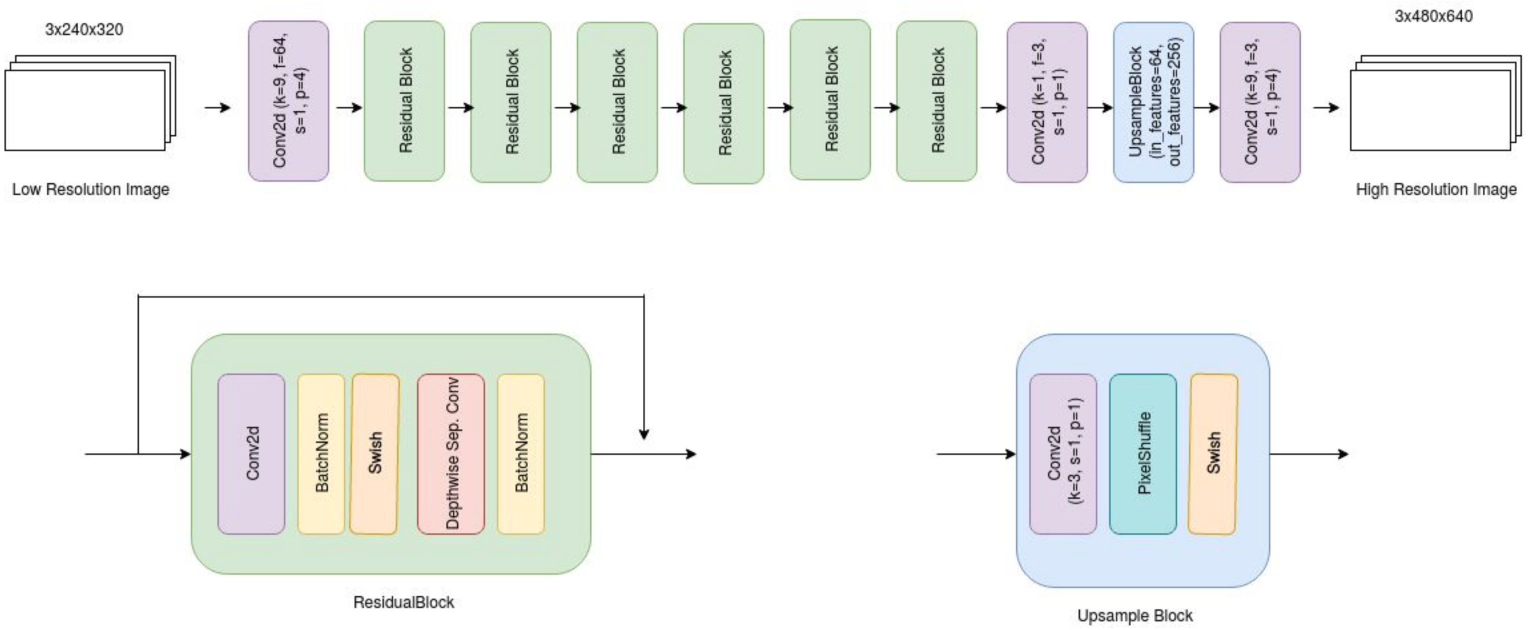


Figure 19: SRGAN Architecture

Results for Super Resolution Task

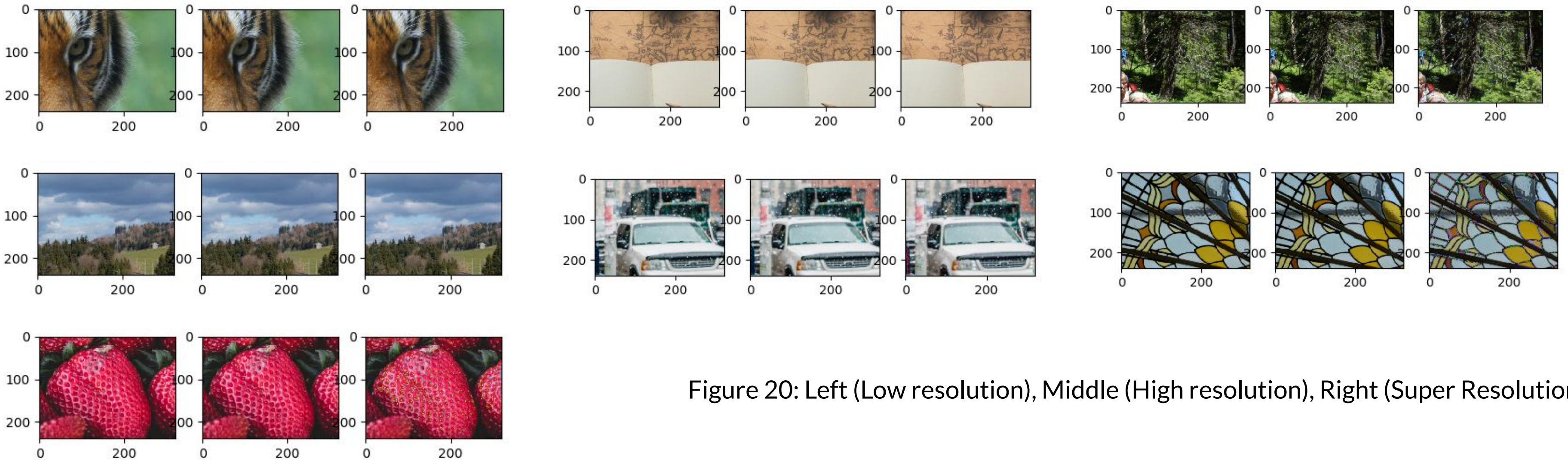


Figure 20: Left (Low resolution), Middle (High resolution), Right (Super Resolution)

Results for Super Resolution Task

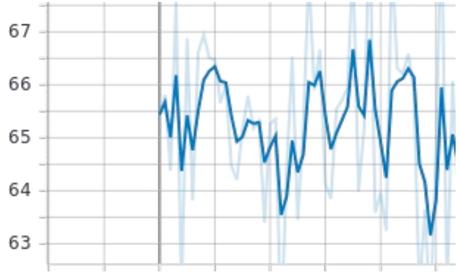


Figure 21:
PSNR(dB)

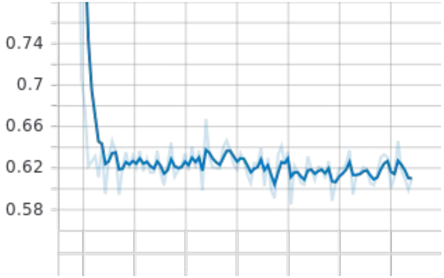


Figure. 22. Discriminator Loss

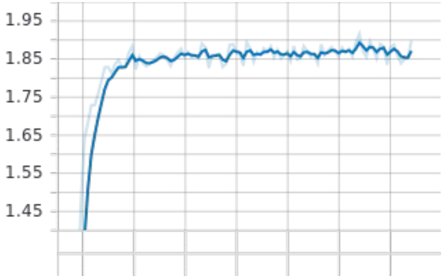


Figure. 23. Generator Adversarial Loss

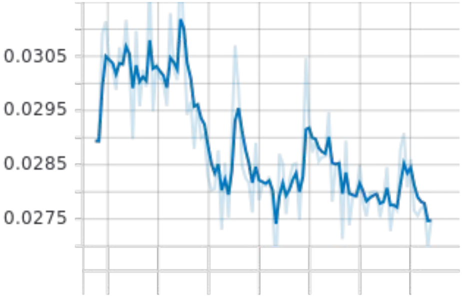


Figure. 24. Generator Content Loss

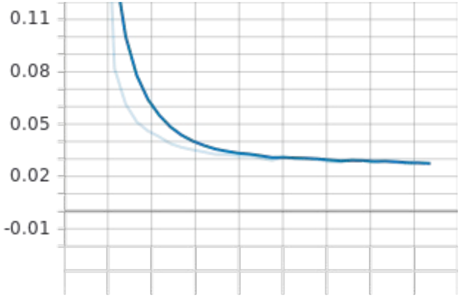


Figure. 25. Generator MSE Loss

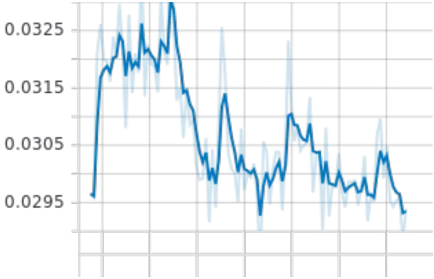


Figure. 26. Generator Total Loss

* x axis = epochs

BGRSRGAN : A Novel Approach

- Based on the results of the previous models, an attempt to address both background replacement and super resolution in an end-to-end trainable framework.
- Changes/contributions made to both BGR and SR architectures:
  - Swish Activation
  - Weight standardization, Grouped Convolutions, Depthwise Separable Convolutions
  - Multipart Loss for Background Replacement Task, Super Resolution Task
  - Another approach to create a dataset for this purpose quickly (Green Screen and Chroma Keying)
  - Lightweight network with real-time inference

Trainable Parameters	Forward/Backward pass size (Mb)	Batch Size	Training time on g4dn.4xlarge (mins/epoch)	Best PSNR (dB)
17,192,908	2556.45	32 (pretrain), 16(trained)	~4	73

Table 3: Model Summary

What we did?

- Dataset
  - Green screen videos with video call-esque situations and background images
  - 600+ green screen video frames, ~12 background images => combinations of upto 7k images
  - Dataloader uses green screen extracted frame and an “original background” to give input image, and a target background as inputs to the model at lower resolution (320x240), output to the model is green screen extracted foreground onto the target background at a higher resolution (640x480)

Figure 27: Samples of Green Screen Dataset

Model Architecture

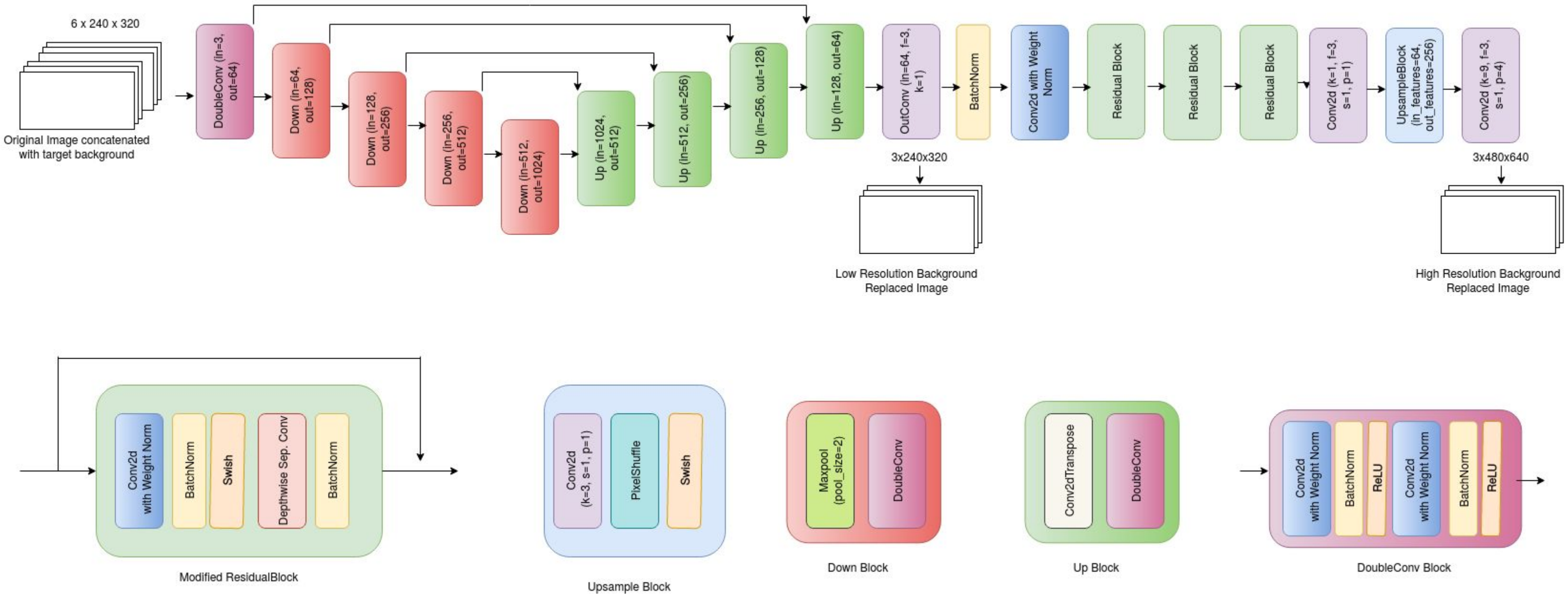
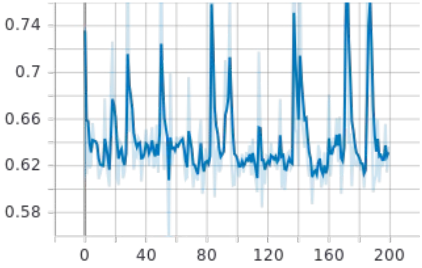
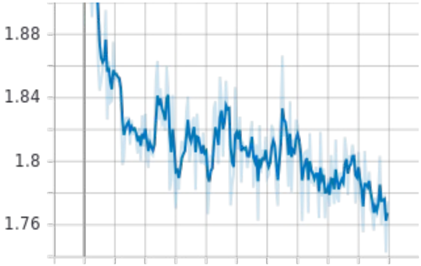


Figure 28: BGRSRGAN Architecture

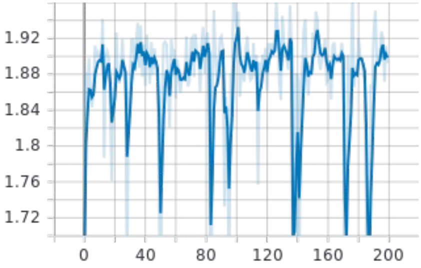
Results - Training 1 (batch_size = 8(Pretrain), 4)



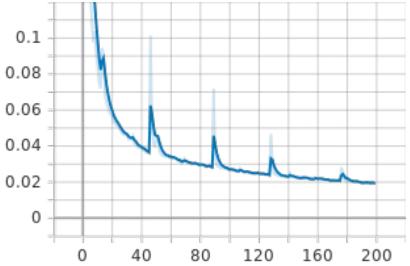
Discriminator Loss



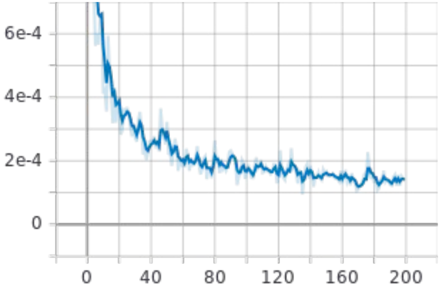
Background Replacement Loss



Generator Adversarial Loss



Generator Content Loss



Generator Total Loss

* x axis = epochs

Results - Training 1 (batch_size = 8(Pretrain), 4)

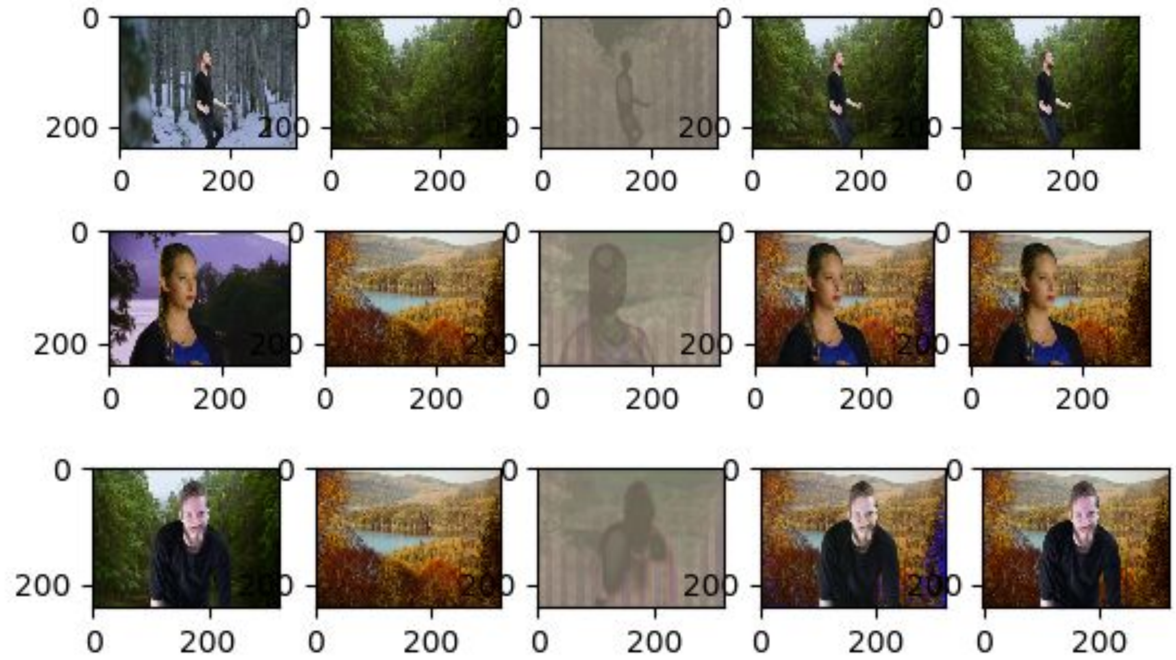
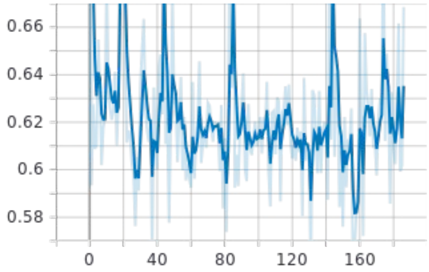
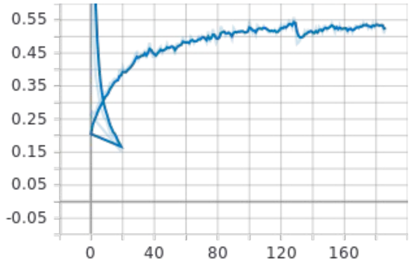


Figure 34: In order (Left to Right) : Original Input Image, Target Background Image, Intermediate Conv output for BG Replacement, BGRSR Output, Target Image

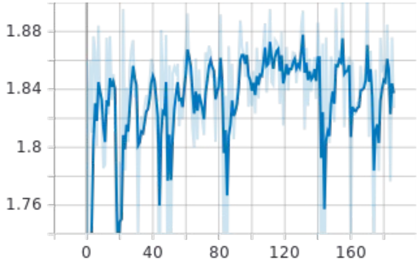
Results - Training n (batch_size = 32(Pretrain), 16)



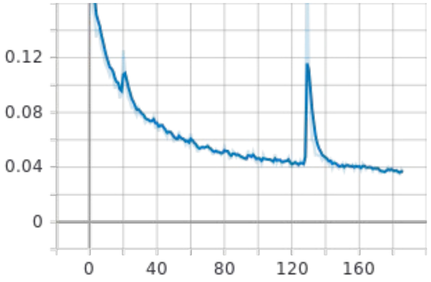
Discriminator Loss



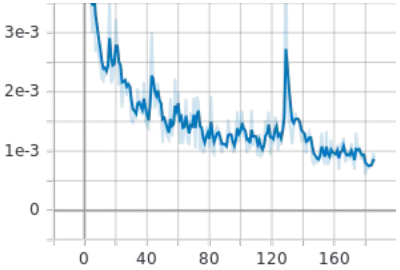
Background Replacement Loss



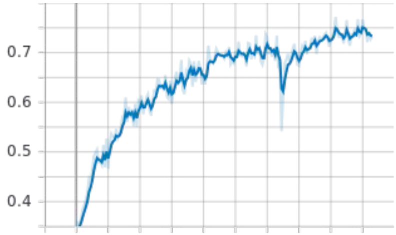
Generator Adversarial Loss



Generator Content Loss



Generator Total Loss



SSIM (best 73)

* x axis = epochs

Promising Results!

Figure 41: In order (Left to Right) : Original Input Image, Target Background Image, Intermediate Conv output for BG Replacement, BGRSR Output, Target Image

How real-time is real-time?

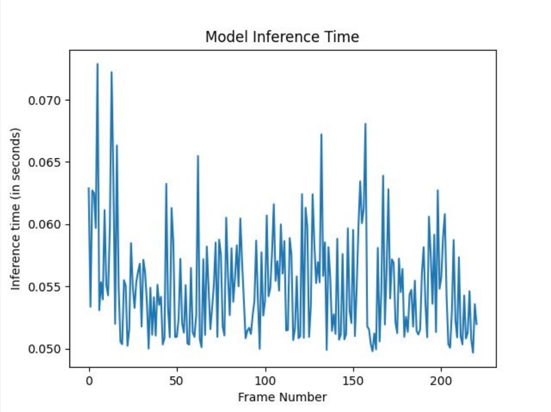


Figure 42: Model Inference Time

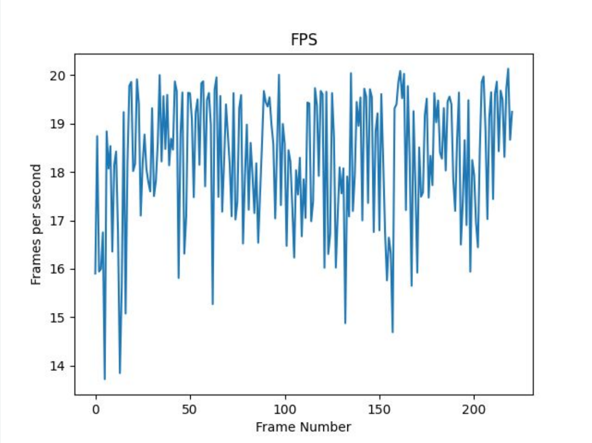
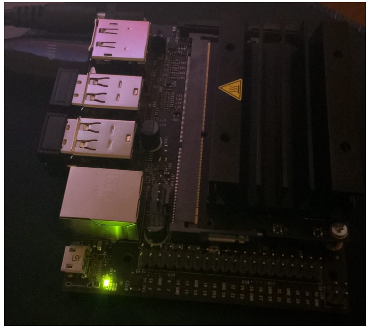


Figure 43: FPS

105 FPS on 8GB GPU

20 FPS on Jetson Nanov1

~0.05 seconds/frame



Failure Cases / Where we can improve?

- In video acquisition, due to motion blur there were discoloration issues which came up which weren’t present in static/single image background replacement + super resolution. This could be resolved by doing some pre-processing and motion blur reduction for frames from videos.
- The green screen approach worked well for this particular task but in order to create a more robust model, would need diverse data with high quality labeling for building model with high SSIM and PSNR metrics.

Conclusions

- The novel approach (BGRSRGAN) was able to achieve good performance as compared to UNET+SRGAN with lesser number of parameters.
- BGRSRGAN has Structural Similarity Index Measure (SSIM) as 73.
- UNET-GAN and SRGAN approach has SSIM as less than 60.
- With enough high quality annotated dataset with diversity, BGRSRGAN would be able to accomplish Background Replacement and Super Resolution in real-time for edge devices

THANK YOU!