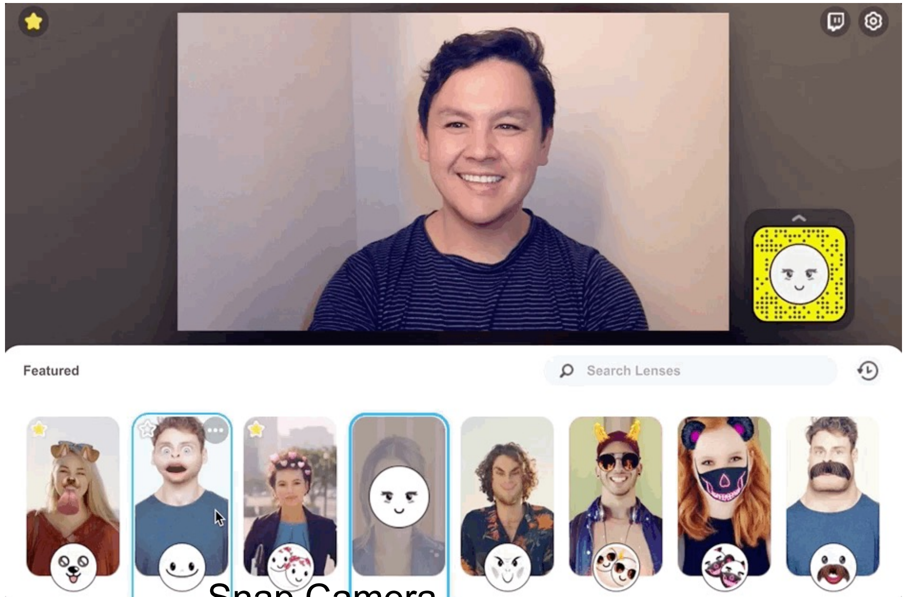
Virtual Try-On Considering Temporal Consistency for Videoconferencing

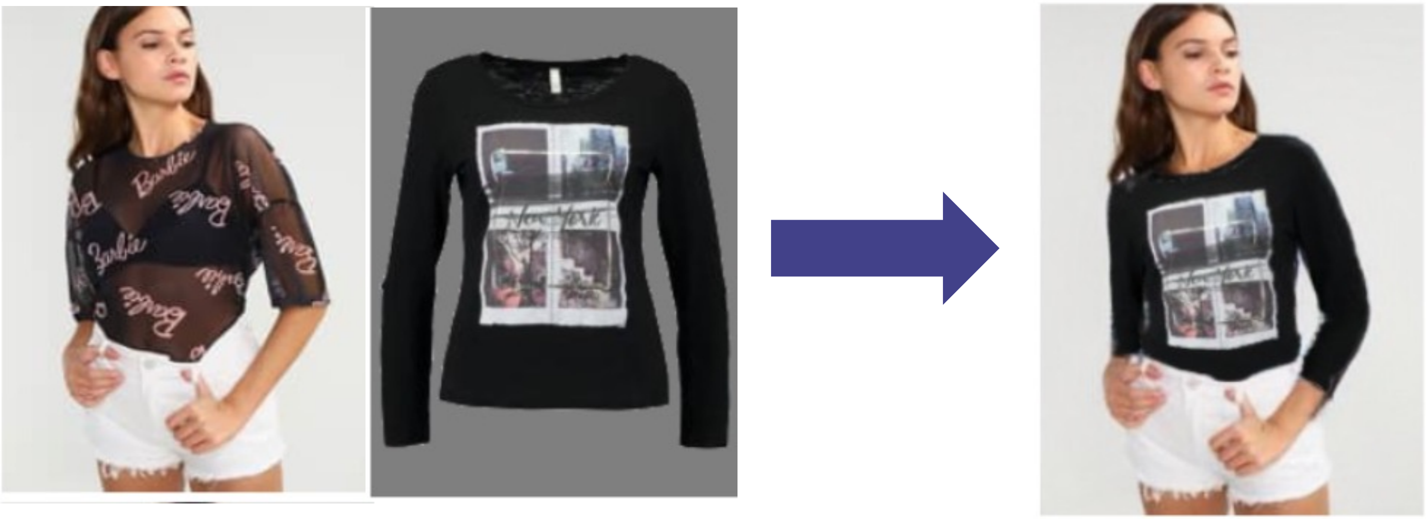
Daiki Shimizu[1], and Keiji Yanai[1]
[1]The University of Electro-Communications, Tokyo

Introduction

- Real-time appearance change in videoconferencing.
  - Style transformation.
  - Virtual backgrounds.
  - Virtual makeup.
  - 3D Avatars.
  - Virtual fitting.

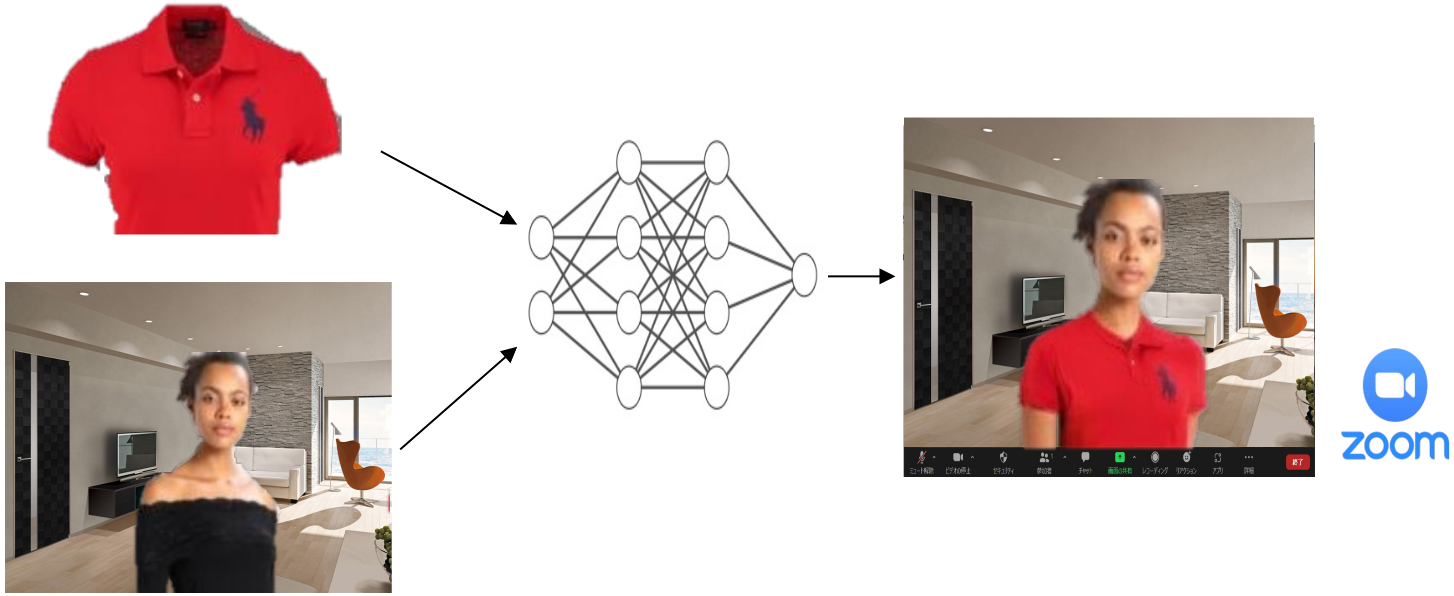


Snap Camera



Objective

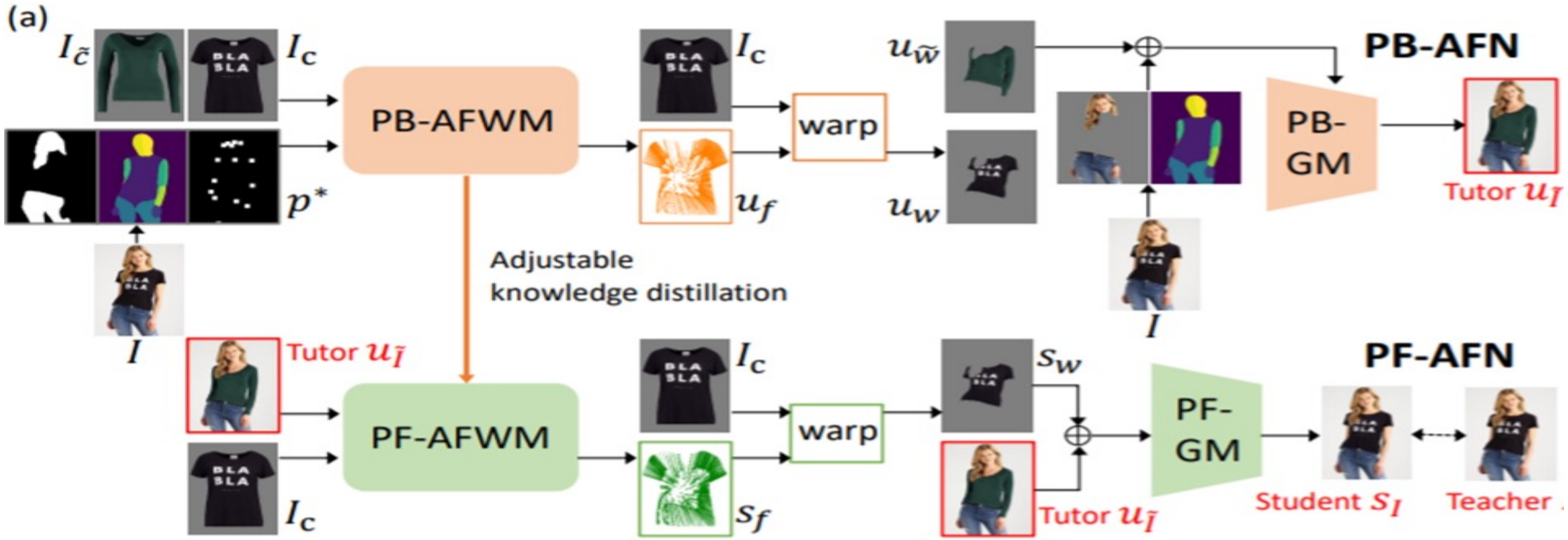
Real-time virtual try-on for videoconferencing considering temporal consistency



Related work - PF-AFN -

[1]Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. Parser-free virtual try-on via distilling appearance fows. In CVPR, 2021.

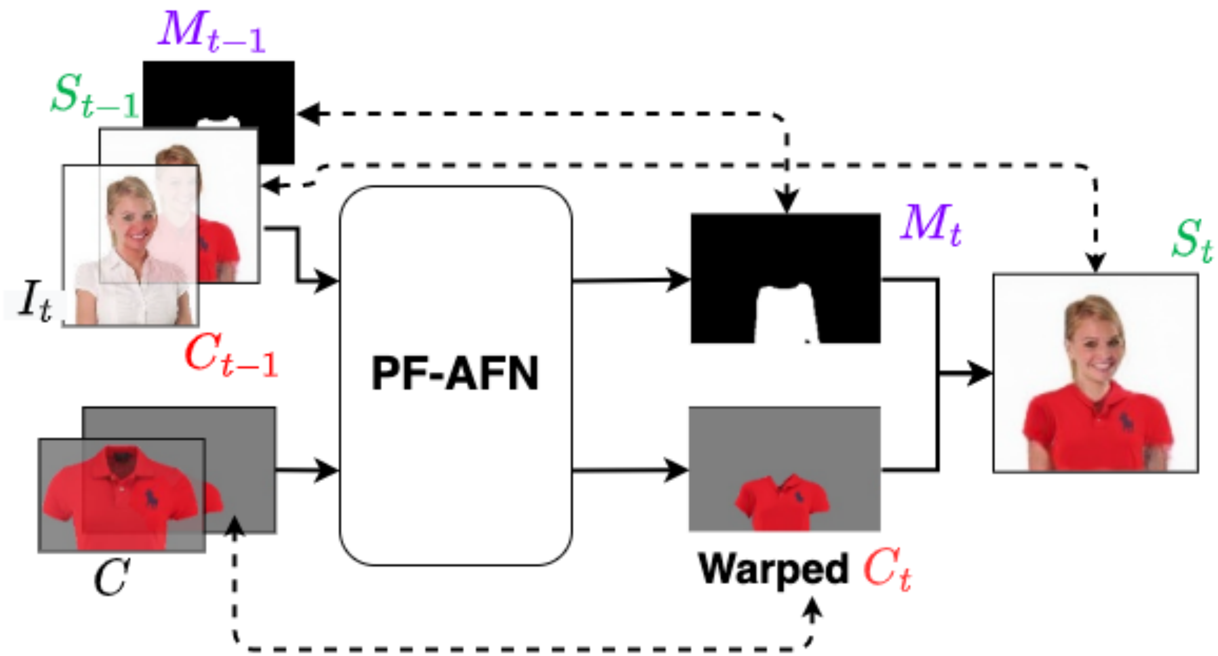
- PF-AFN[1] is a real-time virtual try-on network.
  - – Student model takes only two images as input.
- This work does not take into account temporal information.

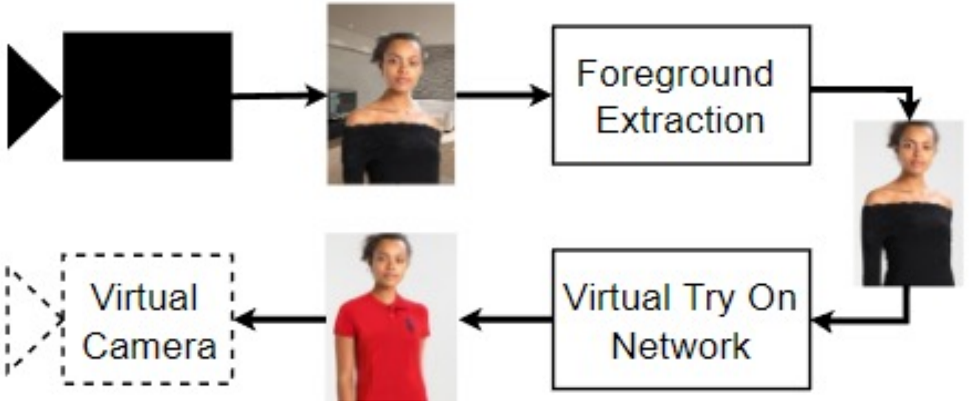


We extend PF-AFN for real-time video virtual try-on.

Contribution

- Propose a method to learn a virtual try-on network considering time-consistency.
- Develop virtual try-on system using virtual camera.

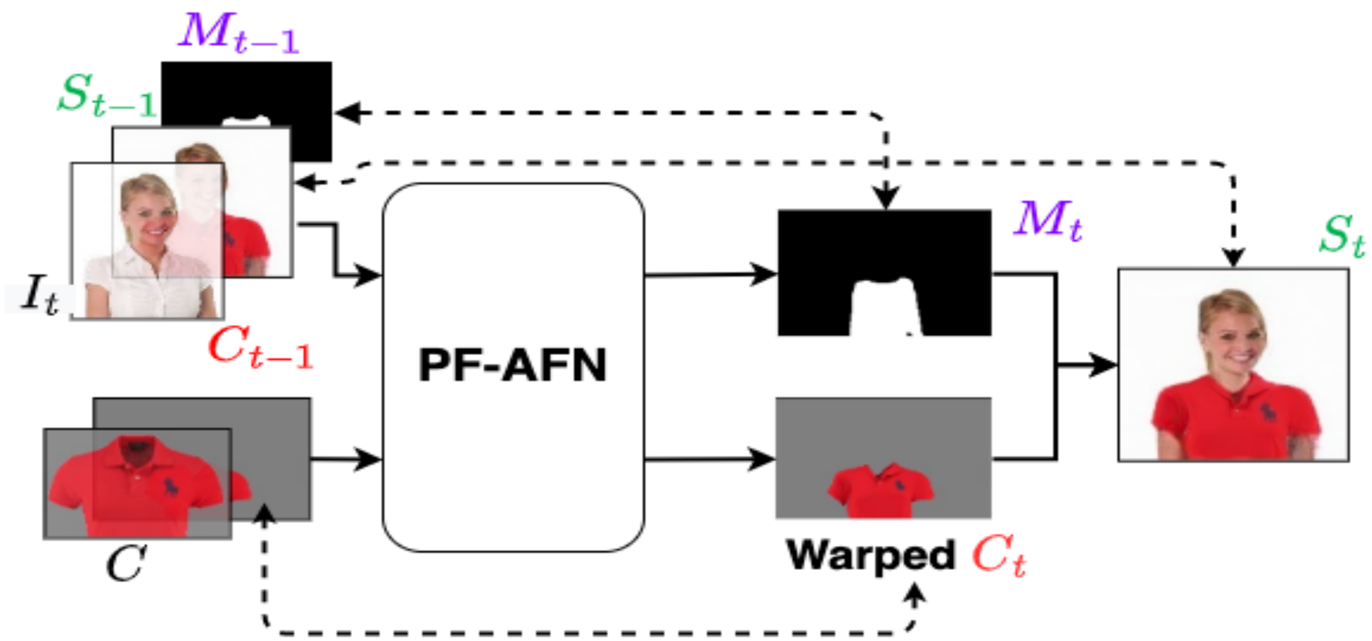




Proposed method

- Base model: **PF-AFN**
- Model inputs:
  - Current image.
  - Template cloth.
  - Previous generated images. (additional input)
- Additional loss:

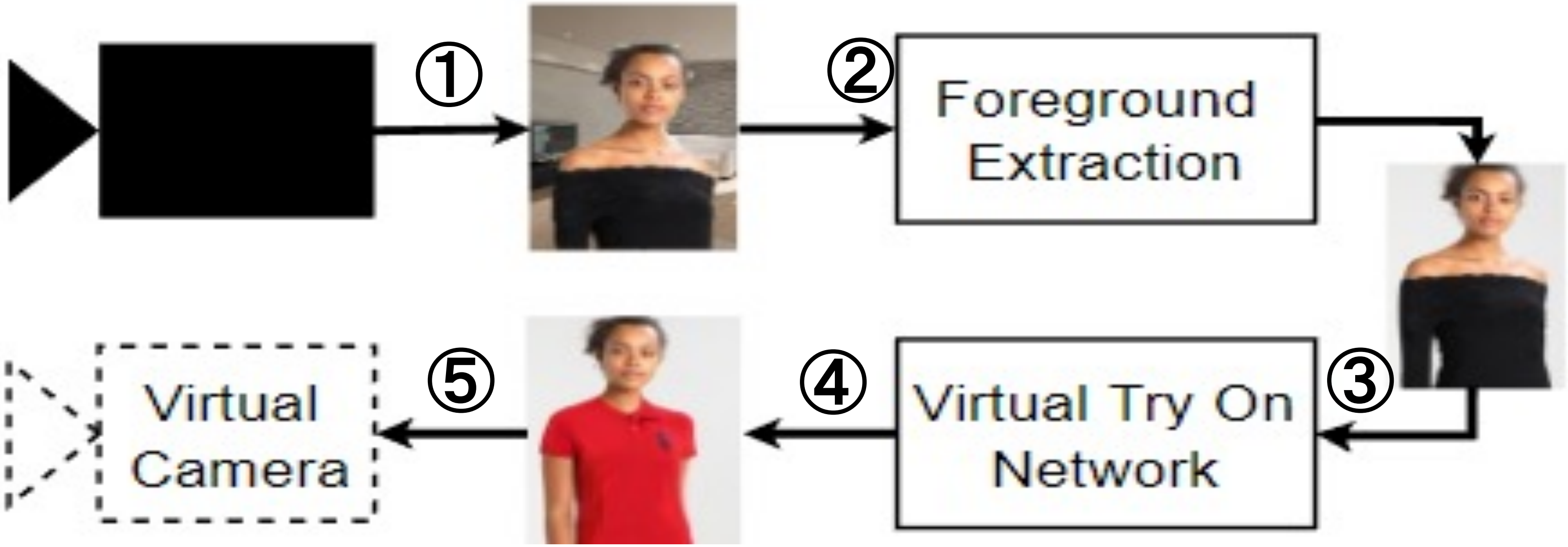
  temporal consistency loss $\mathcal{L}_t$



$$\mathcal{L}_t = \lambda_t(\lambda_{p_1}\mathcal{L}_p(S_t, S_{t-1}) + \lambda_i\mathcal{L}_{L1}(S_t, S_{t-1}) + \lambda_{P_2}\mathcal{L}_P(C_t, C_{t-1}) + \lambda_c\mathcal{L}_{L1}(C_t, C_{t-1}) + \lambda_M\mathcal{L}_{L1}(M_t, M_{t-1}))$$

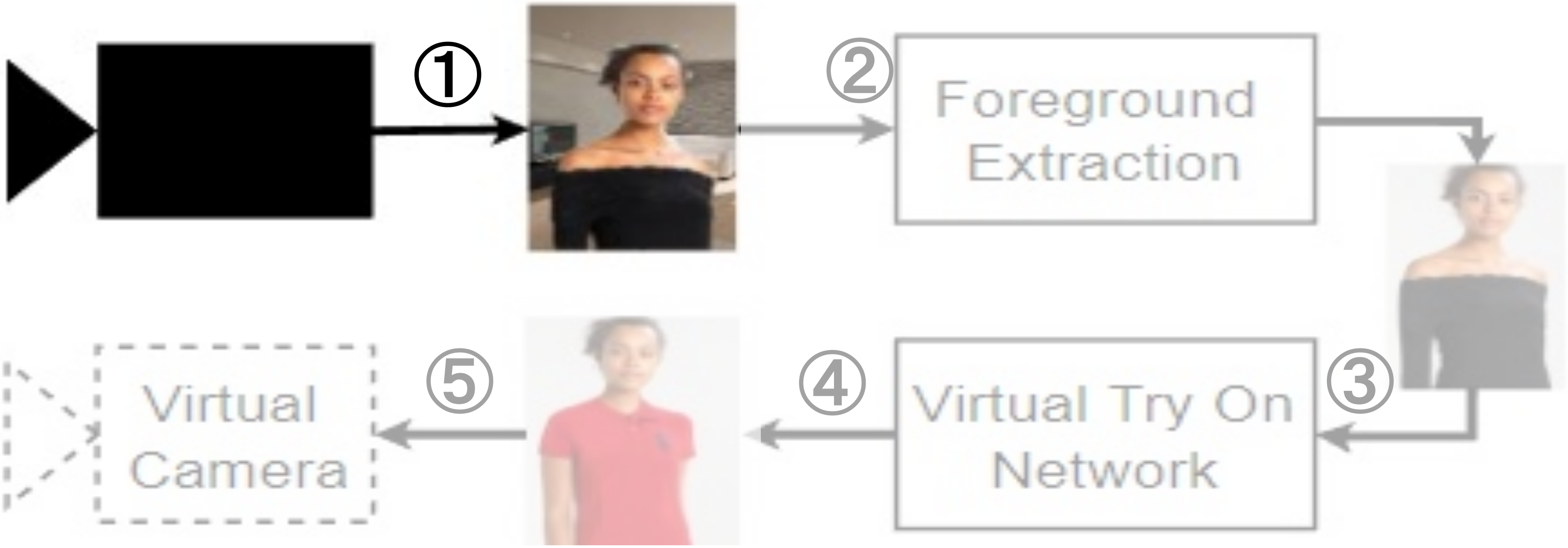
System architecture

1. Take one frame from the camera stream.
2. Remove background using a pre-trained segmentation model.
3. Provide the foreground image to the proposed try-on model.
4. Obtain the cloth-changed image from the model.
5. Provide it to the virtual camera.



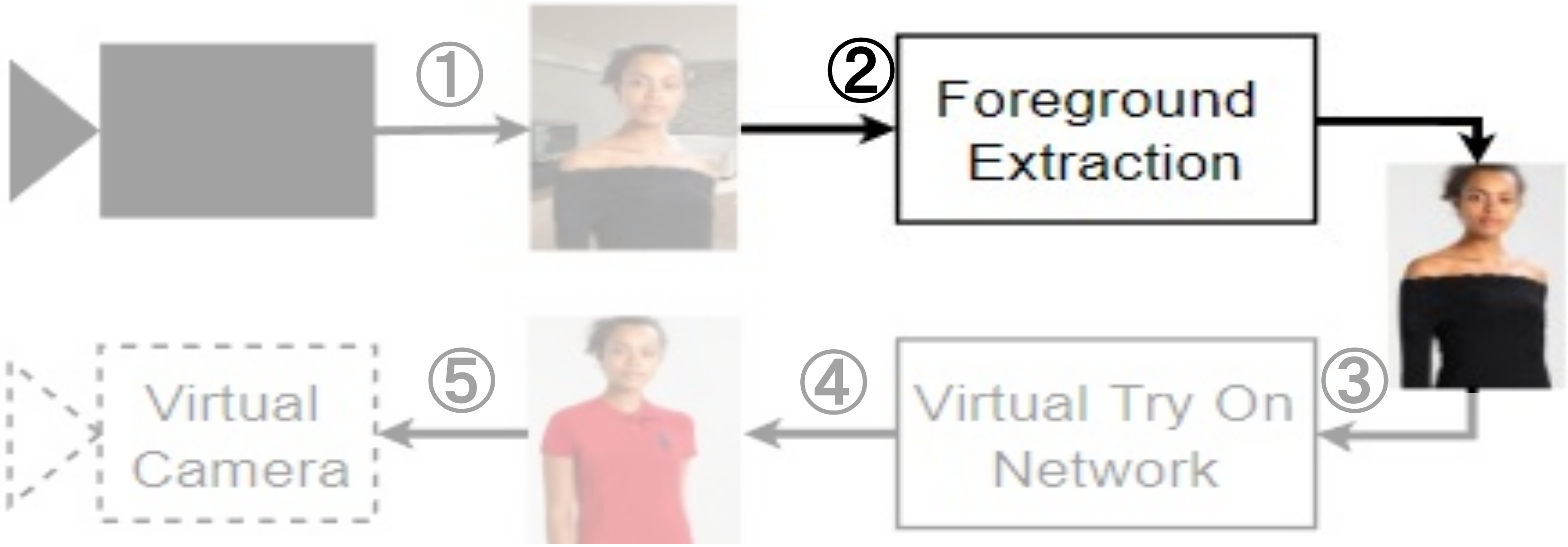
System architecture

1. Take one frame from the camera stream.
2. Remove background using a pre-trained segmentation model.
3. Provide the foreground image to the proposed try-on model.
4. Obtain the cloth-changed image from the model.
5. Provide it to the virtual camera.



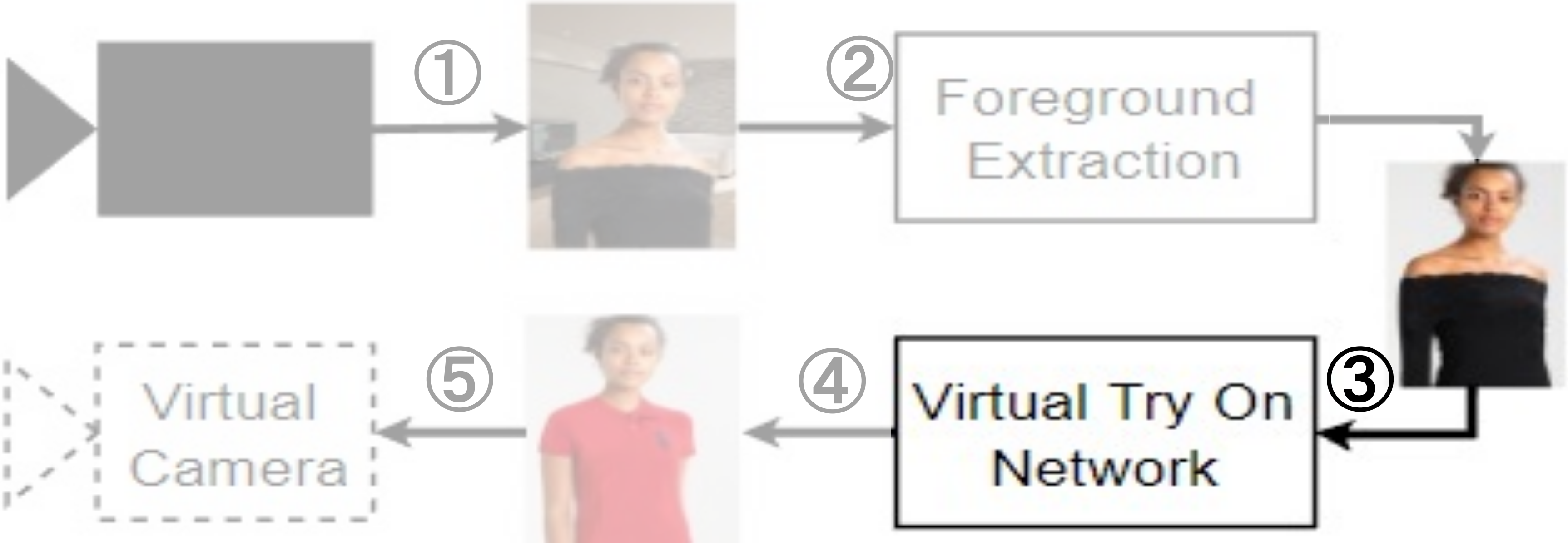
System architecture

1. Take one frame from the camera stream.
2. Remove background using a pre-trained segmentation model.
3. Provide the foreground image to the proposed try-on model.
4. Obtain the cloth-changed image from the model.
5. Provide it to the virtual camera.



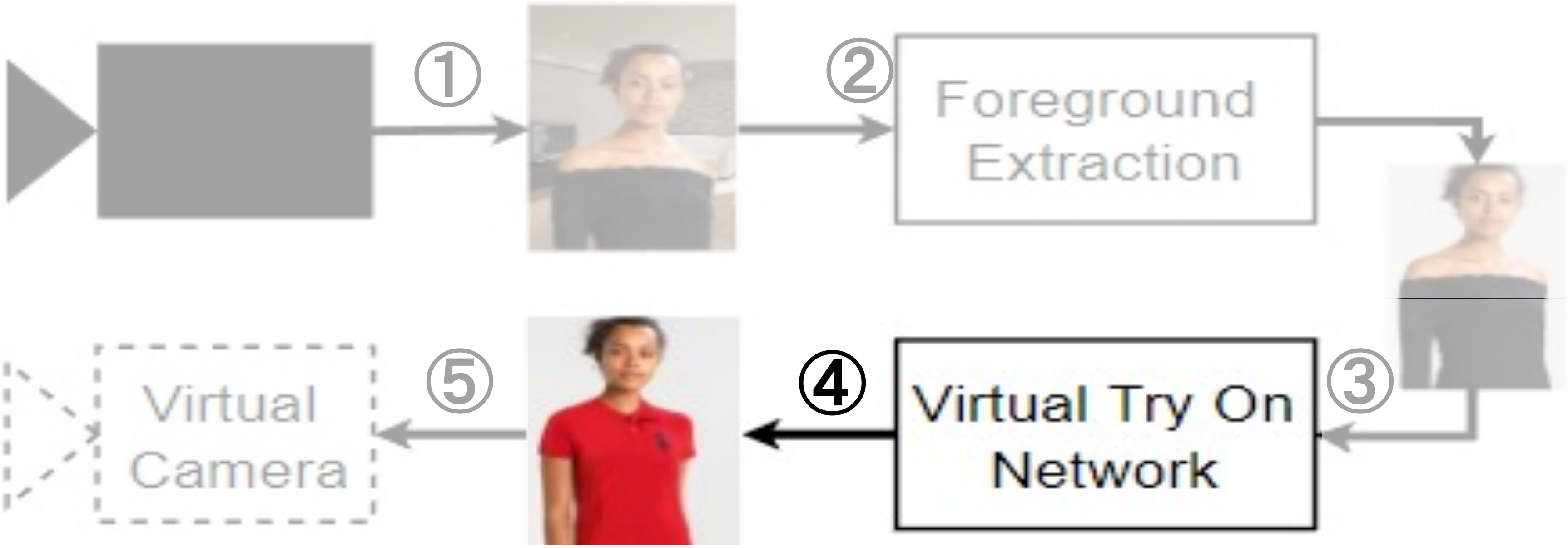
System architecture

1. Take one frame from the camera stream.
2. Remove background using a pre-trained segmentation model.
3. Provide the foreground image to the proposed try-on model.
4. Obtain the cloth-changed image from the model.
5. Provide it to the virtual camera.



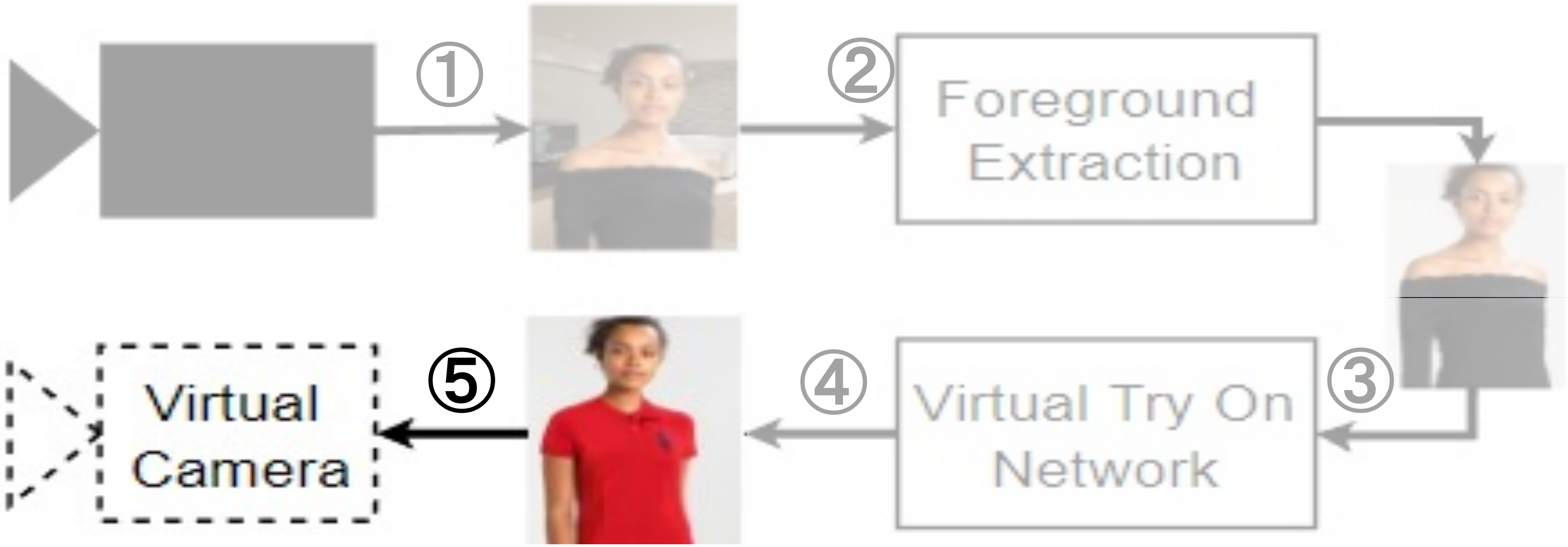
System architecture

1. Take one frame from the camera stream.
2. Remove background using a pre-trained segmentation model.
3. Provide the foreground image to the proposed try-on model.
4. Obtain the cloth-changed image from the model.
5. Provide it to the virtual camera.

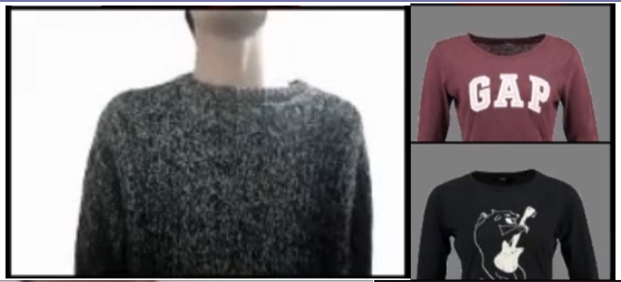


System architecture

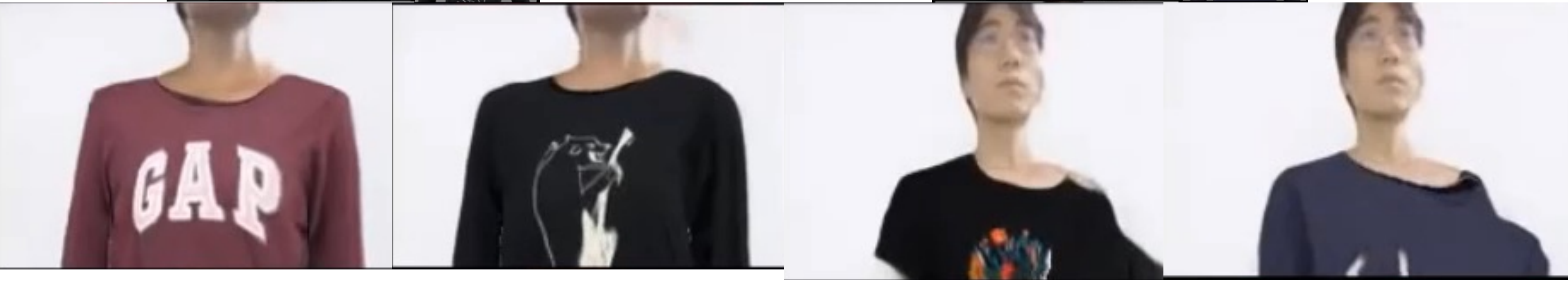
1. Take one frame from the camera stream.
2. Remove background using a pre-trained segmentation model.
3. Provide the foreground image to the proposed try-on model.
4. Obtain the cloth-changed image from the model.
5. Provide it to the virtual camera.



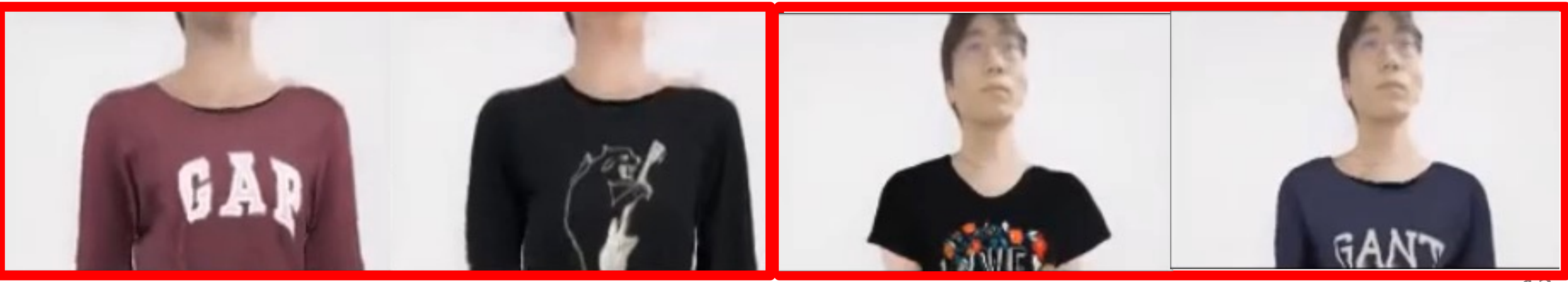
Demonstration



PF-AFN

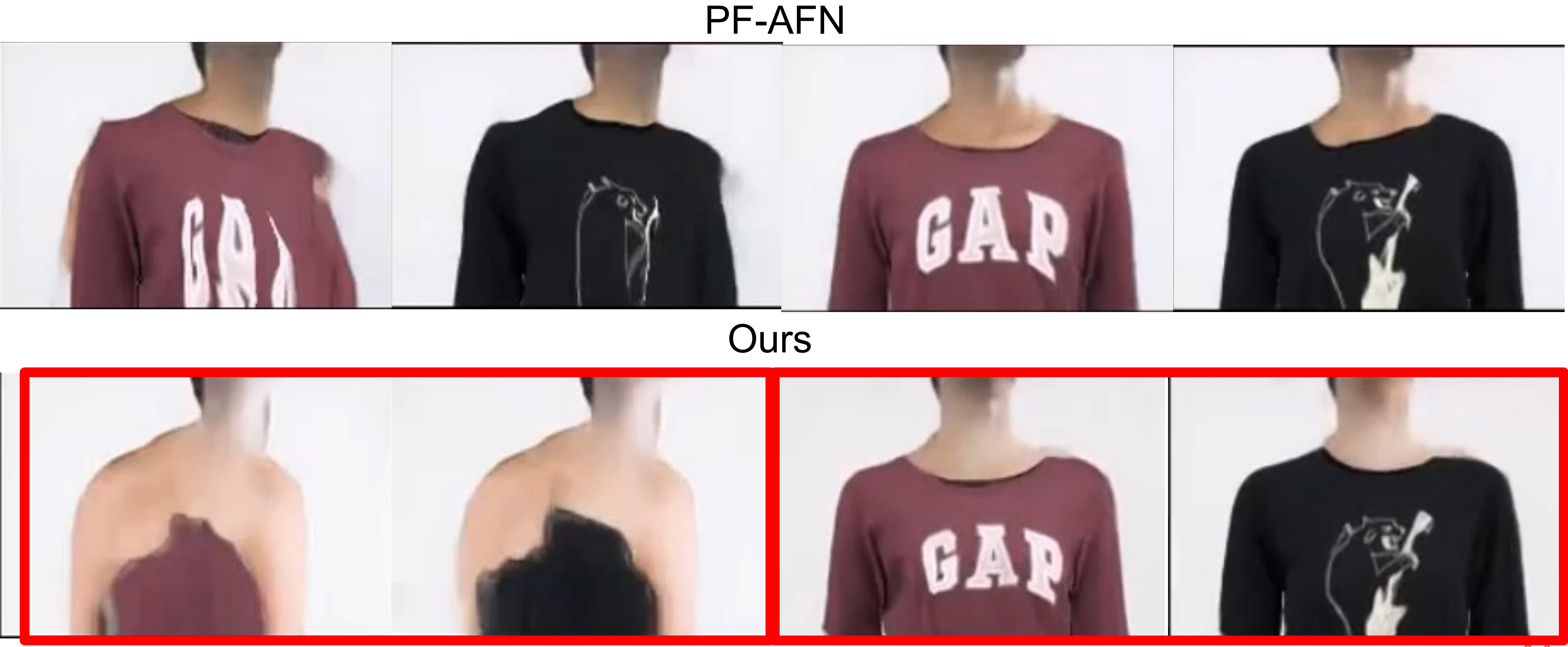


Ours



Limitation

- Difficult to correct failed frame like initial frame or quick movement.



PF-AFN

Ours

Conclusion

- Extended PF-AFN by adding temporal consistency loss.
  - Suppressed the flicker to some extent.
  - Still difficult to:
    - Respond to quick movements.
    - Correct failed pixels.

- Created a virtual fitting system for videoconferencing.

UEC
TOKYO