machine learning conference

Virtual Try-On Video: Joint Dress and Pose Transfer for Fashion Animation

Ilia Vitsnudel @vitsnude
Principal Scientist | Lab126

Liza Potikha @lizap
Applied Researcher | Lab126

Virtual Try-On: Problem Formulation

User Image | Catalog Image | Pose Constraints

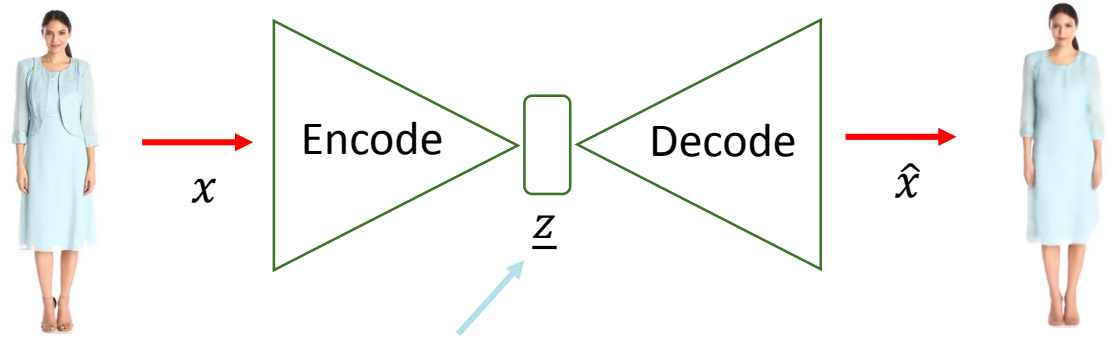


Result Image

- Change Dress and Pose
- Holistic Approach:
  - Using RGB images only
  - Train system by presenting large pool of images with no explicit model

Compact Representation of Outfits – Latent Space

- Construct a system that can represent people wearing outfits

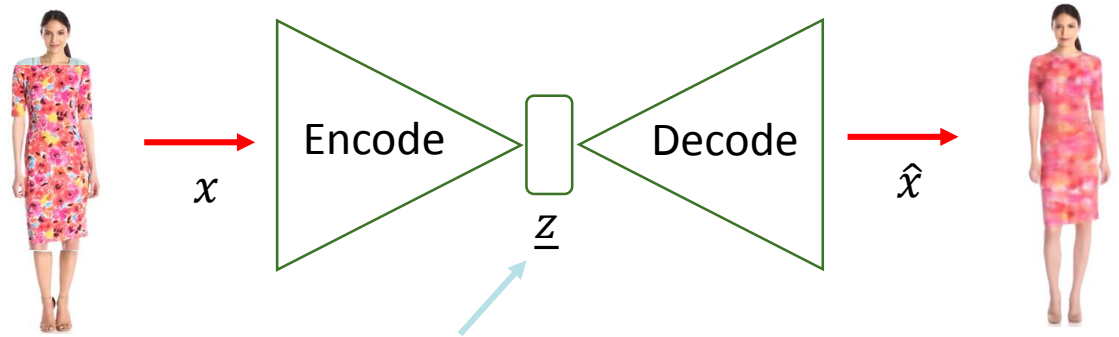


Latent Vector $\underline{z}$ – representing people wearing outfits

- The output is a smoothed version of the input image
- GAN adds details, but can cause artifacts

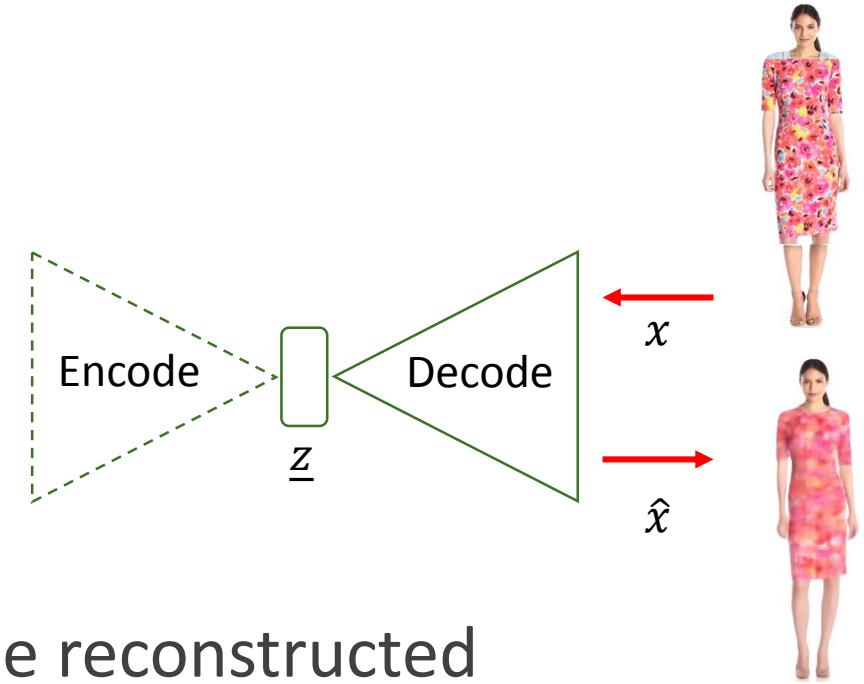
Forward Procedure to Fix Outfit

- Apply this system to fix the outfit



Latent Vector $\underline{z}$ – representing people wearing outfits

Iterative Backward Procedure to Fix Output



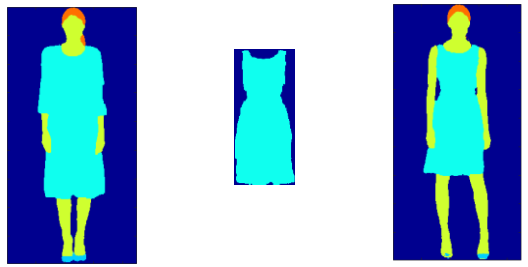
- x – Image to be reconstructed
- Present x at the output of the decoder (generator)
- Glide back through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward procedure to get a reconstruction $\hat{x}$.

TripleGuidedCNN for Joint Pose and Dress Transfer

Put I_{Dress} on I_{Input} guided by S_{Output}

(A new segmentation in a new pose).

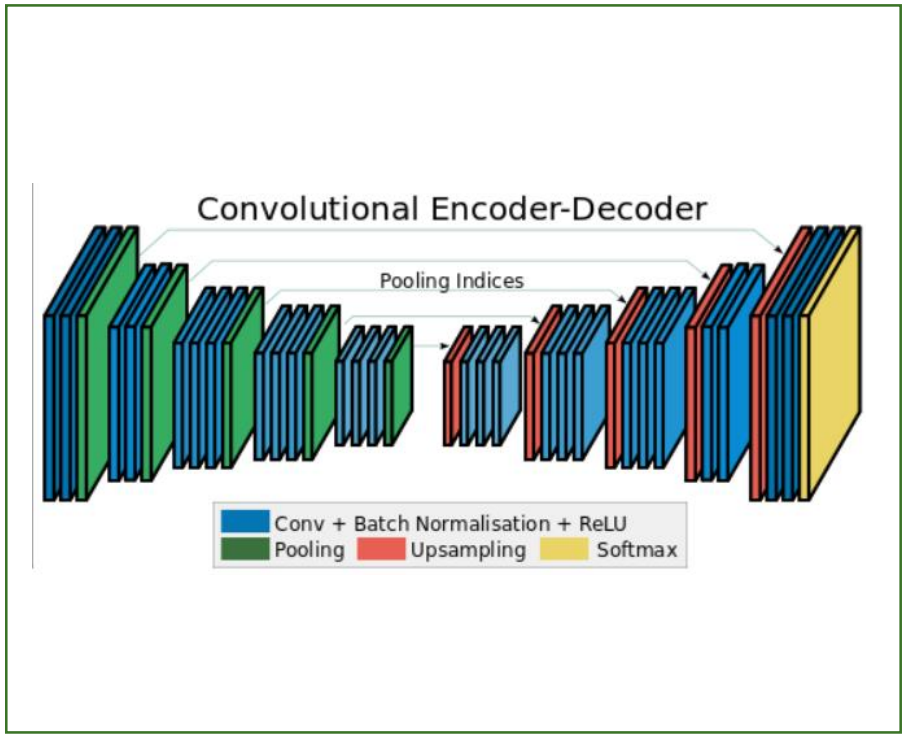
Use S_{Input} and S_{Dress} as additional guidance



S_{Input} / S_{Dress} / S_{Output}

I_{Input}

I_{Dress}



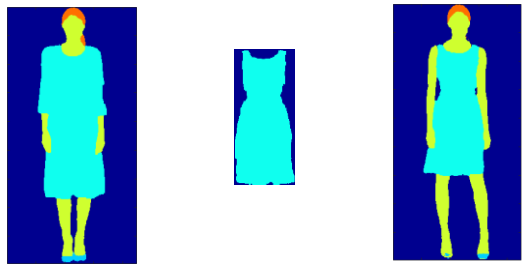
$\hat{I}_{Output}$

TripleGuidedCNN for Joint Pose and Dress Transfer

Put I_{Dress} on I_{Input} guided by S_{Output}

(A new segmentation in a new pose).

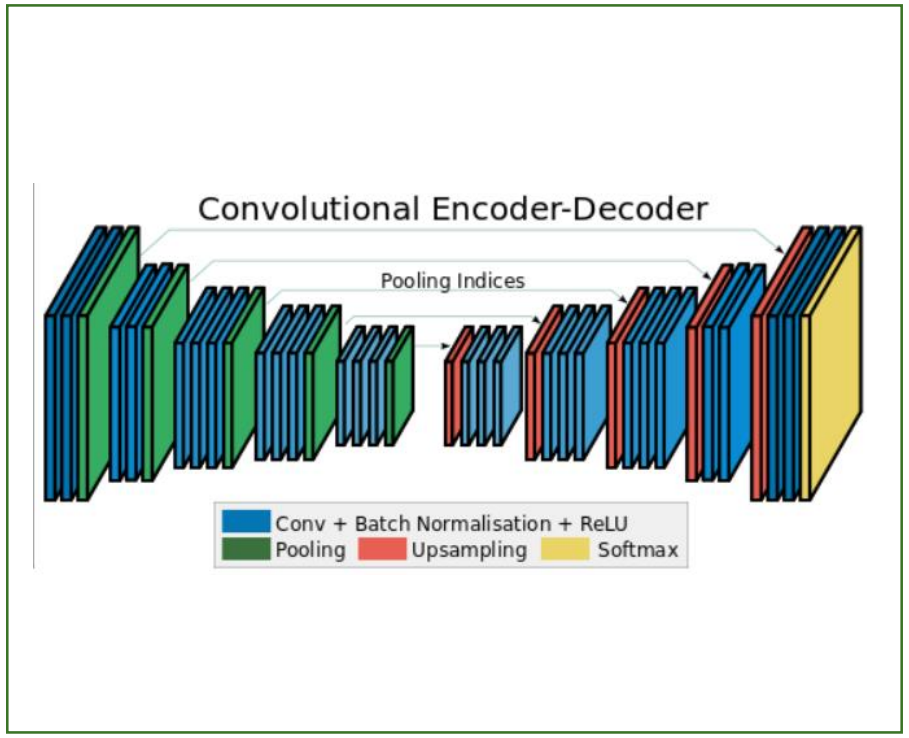
Use S_{Input} and S_{Dress} as additional guidance



S_{Input} / S_{Dress} / S_{Output}

I_{Input}

I_{Dress}



$\hat{I}_{Output}$

We need:

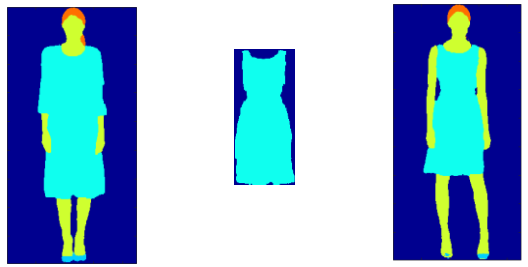
1. A DataSet for Training
2. To Construct Consistent S_{Output}
3. To Address Generative Network Blurriness

TripleGuidedCNN for Joint Pose and Dress Transfer

Put I_{Dress} on I_{Input} guided by S_{Output}

(A new segmentation in a new pose).

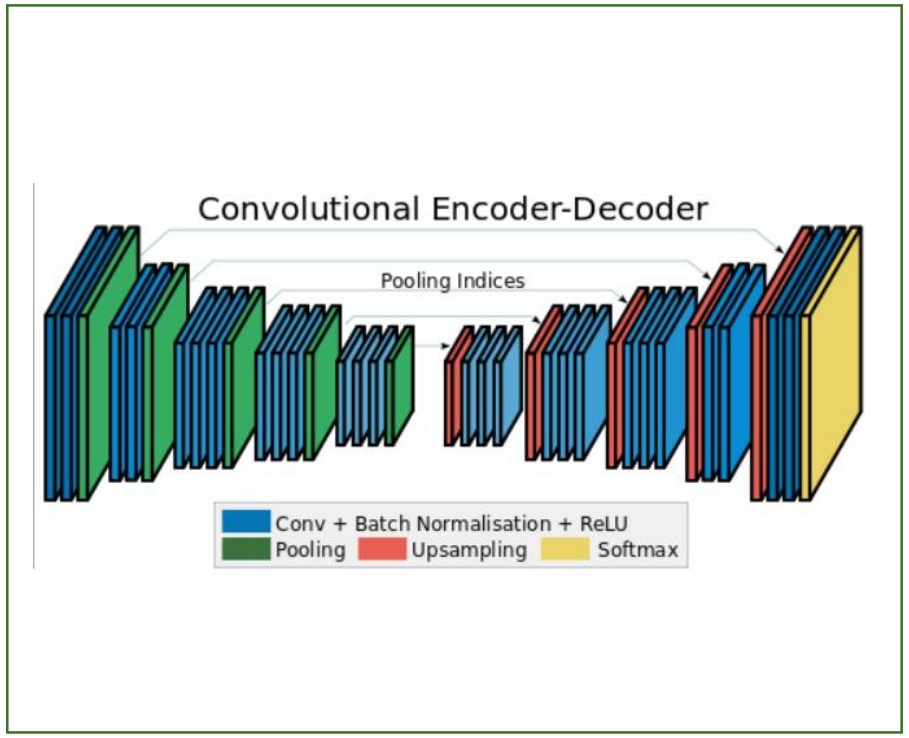
Use S_{Input} and S_{Dress} as additional guidance



S_{Input} / S_{Dress} / S_{Output}

I_{Input}

I_{Dress}



$\hat{I}_{Output}$

We need:

1. A DataSet for Training
2. To Construct Consistent S_{Output}
3. To Address Generative Network Blurriness

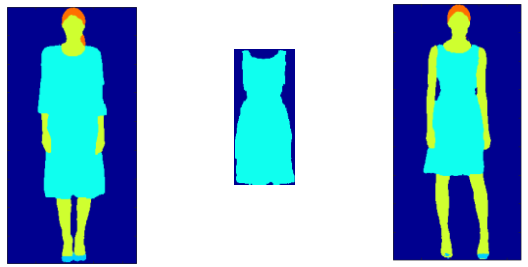
→ Pose and Dress DataSet

TripleGuidedCNN for Joint Pose and Dress Transfer

Put I_{Dress} on I_{Input} guided by S_{Output}

(A new segmentation in a new pose).

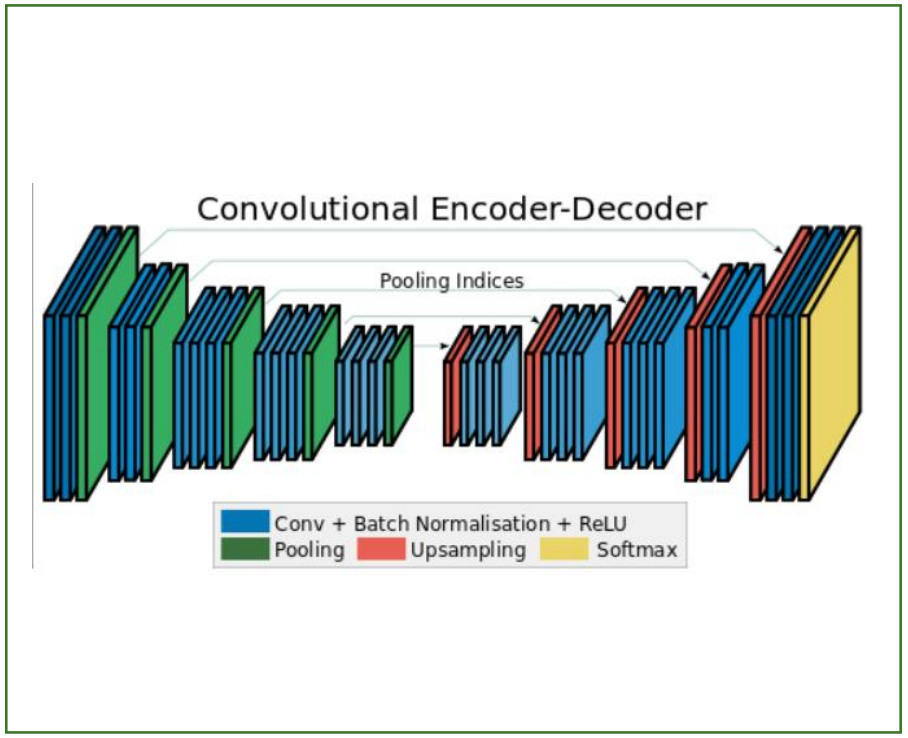
Use S_{Input} and S_{Dress} as additional guidance



S_{Input} / S_{Dress} / S_{Output}

I_{Input}

I_{Dress}



$\hat{I}_{Output}$

We need:

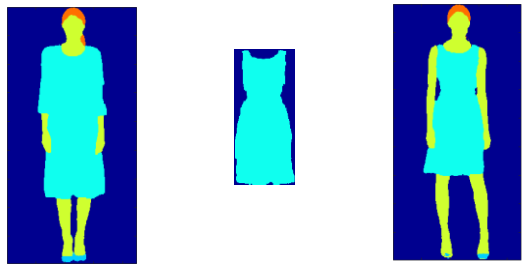
1. A DataSet for Training → Pose and Dress DataSet
2. To Construct Consistent S_{Output} → Create by Backward Working on MultiPathCNN
3. To Address Generative Network Blurriness

TripleGuidedCNN for Joint Pose and Dress Transfer

Put I_{Dress} on I_{Input} guided by S_{Output}

(A new segmentation in a new pose).

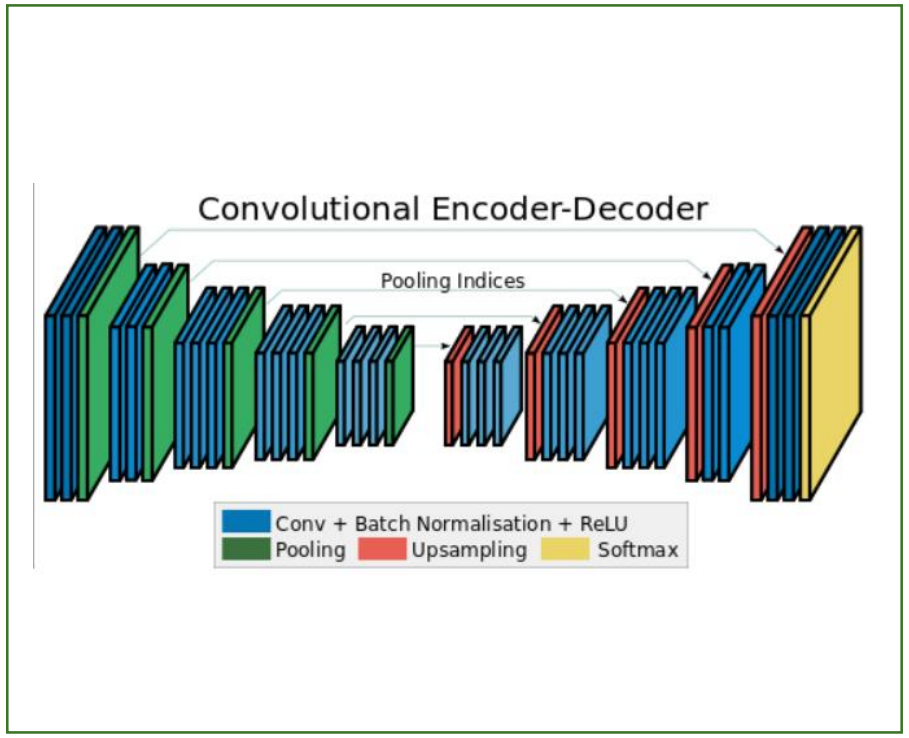
Use S_{Input} and S_{Dress} as additional guidance



S_{Input} / S_{Dress} / S_{Output}

I_{Input}

I_{Dress}



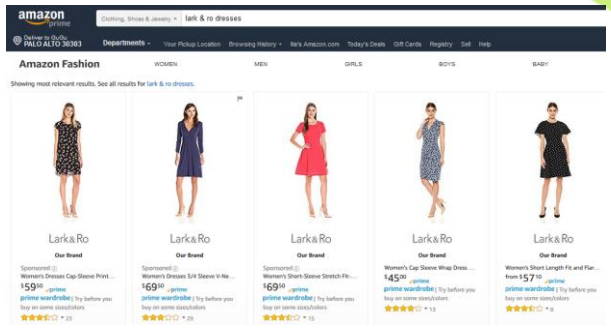
$\hat{I}_{Output}$

We need:

1. A DataSet for Training → Pose and Dress DataSet
2. To Construct Consistent S_{Output} → Create by Backward Working on MultiPathCNN
3. To Address Generative Network Blurriness → ReMap CNN

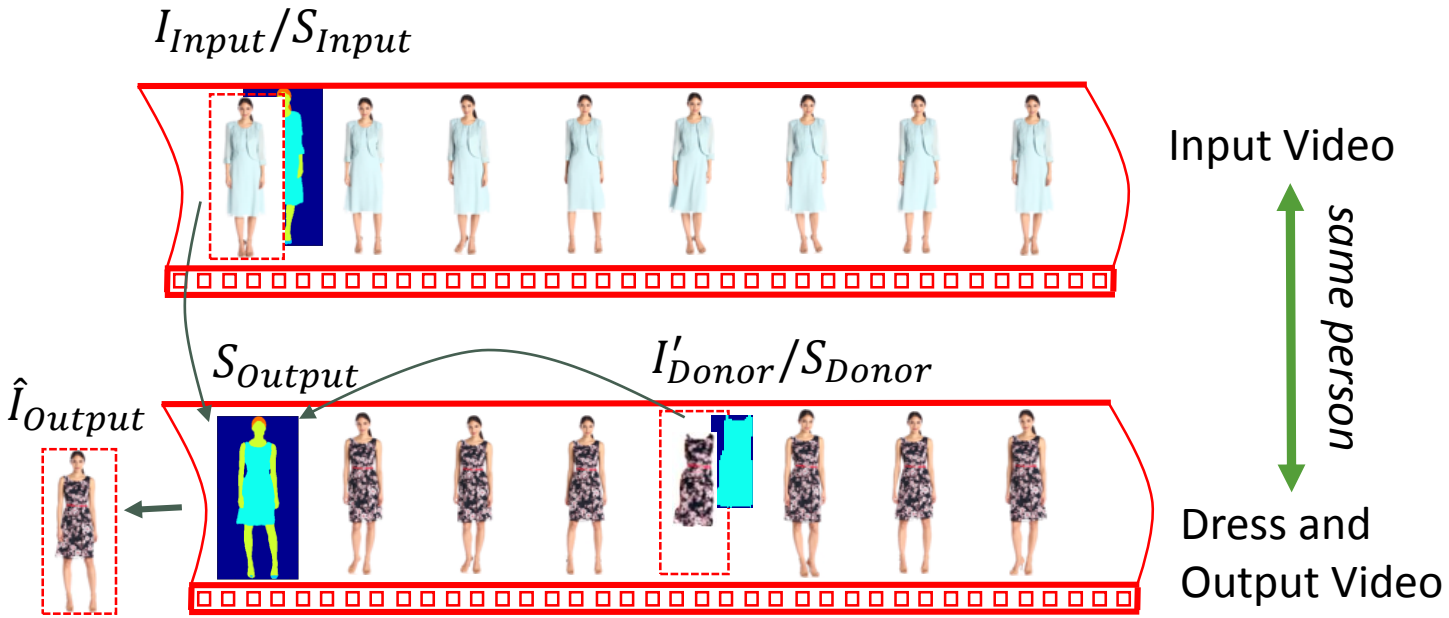
Pose-And-Dress Dataset for Training Joint Pose and Dress Transfer

- Videos from the Amazon Catalog (~5K)
- Frames in different poses extracted from each video
- Same person - clustered using Face Recognition



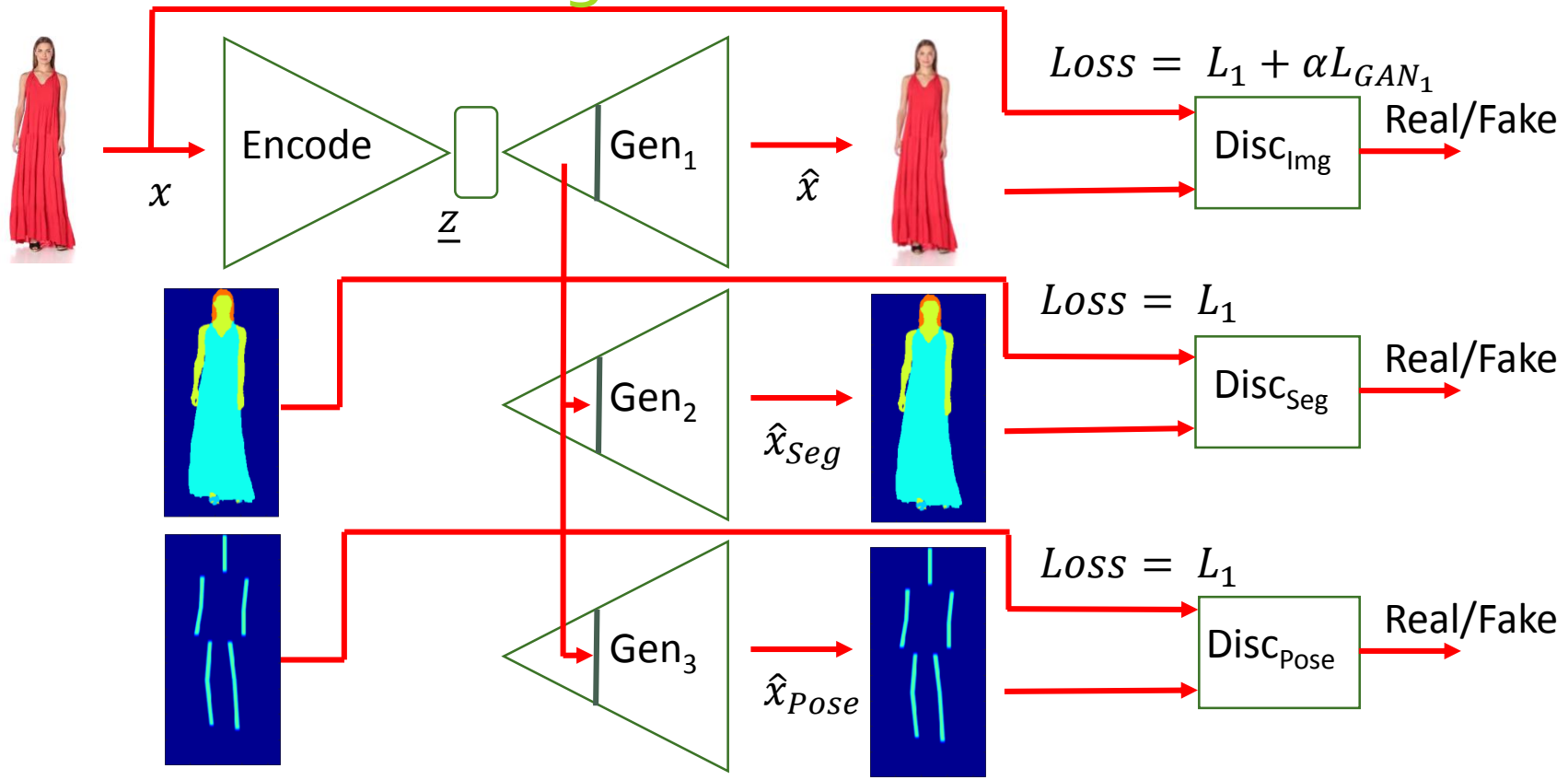
Training sample

- Constructed from two videos of the same person
- I'_{Donor} - is a **dress only** image extracted from donor image
- $\hat{I}_{Output}$ - ground truth of I_{Input} wearing I'_{Donor} dress
- S_{Output} - a target segmentation (new pose)



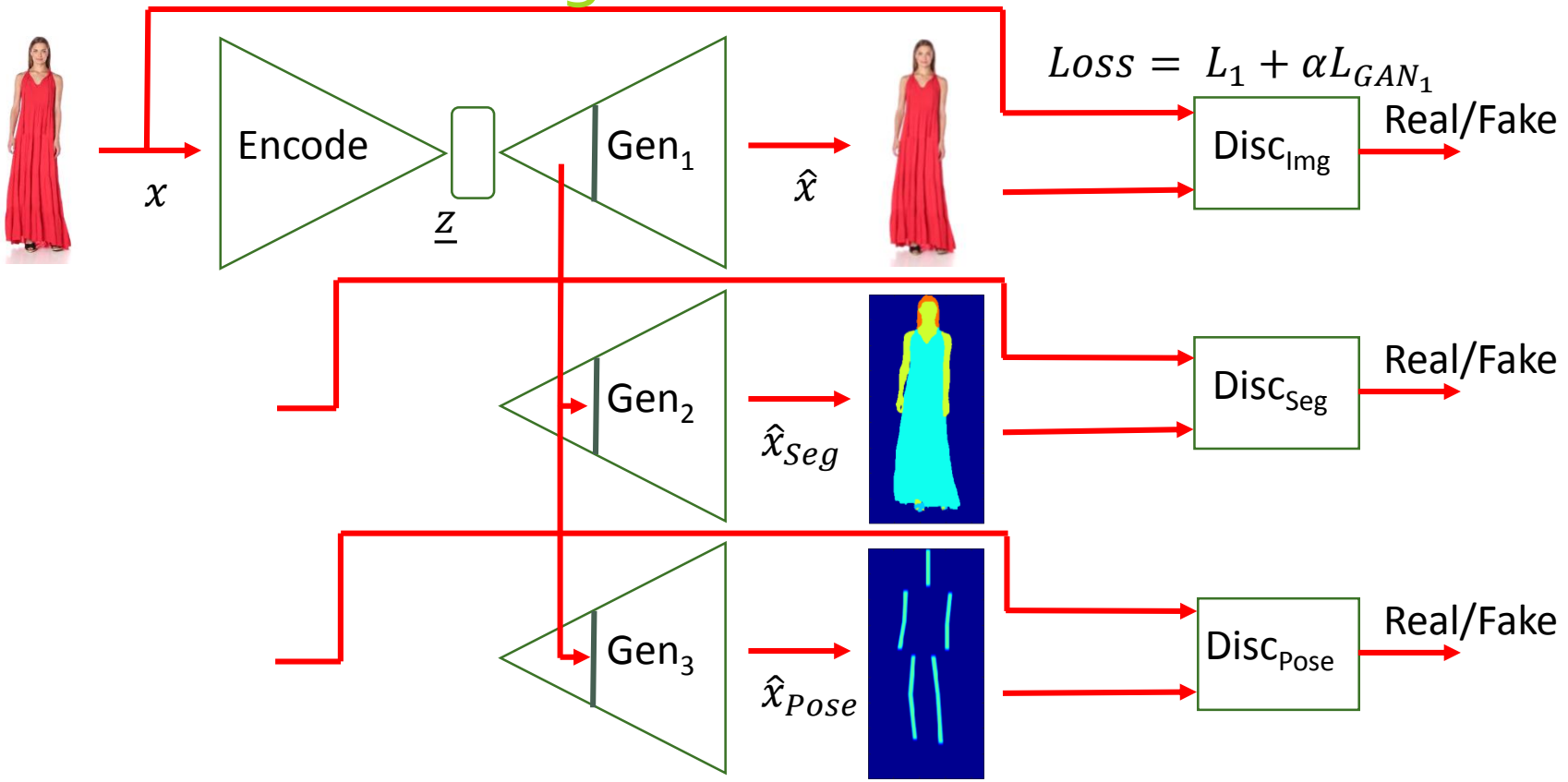
For the **inference time** – we need to construct the target segmentation S_{Output}

MultiPathCNN to Construct Target Segmentation Architecture and Training



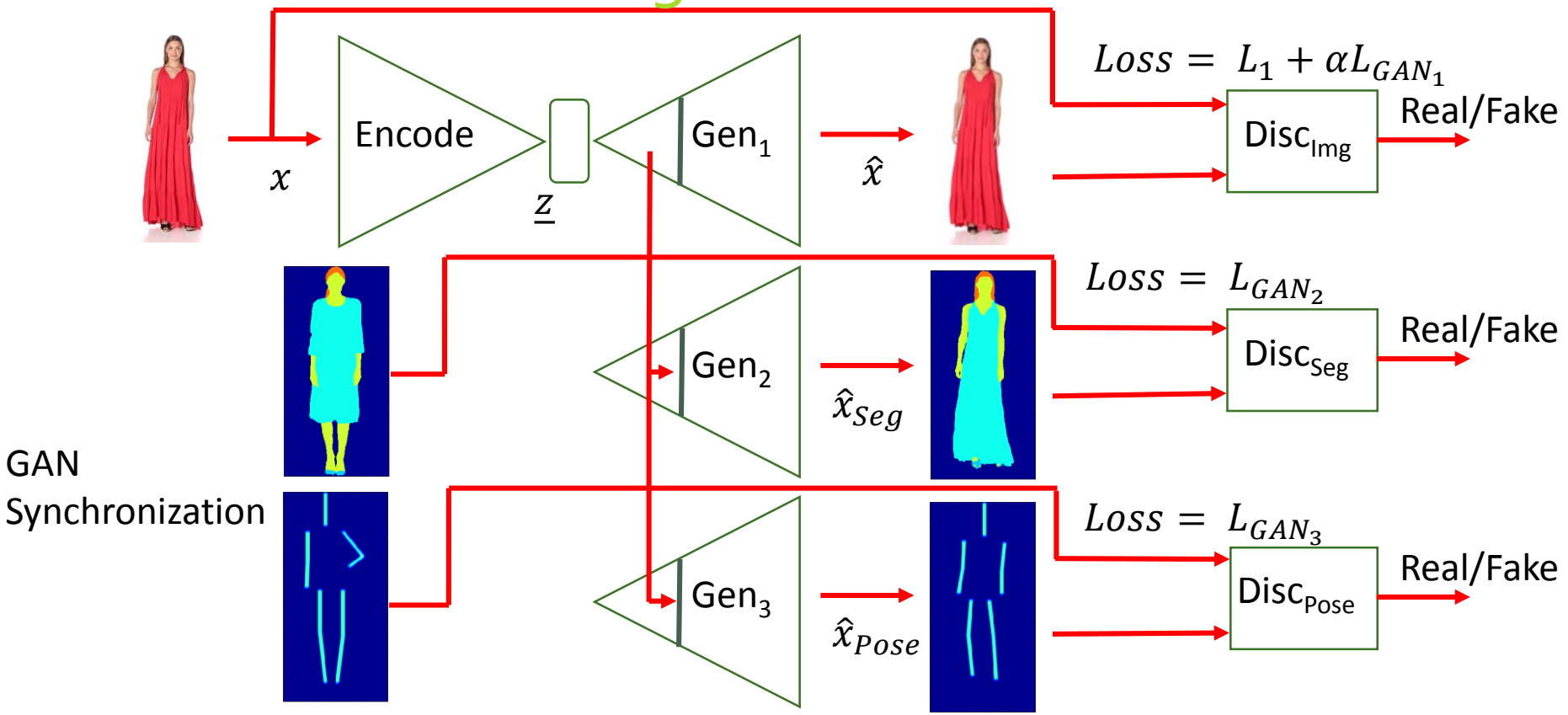
- Auto-encoder system with small latent space vector $\underline{z}$
- Parallel decoder paths (Image, Clothes Segmentation, Pose image) share weights on first layers
- Learns to synchronize during training
- For inference only the input image is shown and the system generates reconstructed human image as well as its appropriate segmentation and pose

MultiPathCNN to Construct Target Segmentation Architecture and Training



- Auto-encoder system with small latent space vector $\underline{z}$
- Parallel decoder paths (Image, Clothes Segmentation, Pose image) share weights on first layers
- Learns to synchronize during training
- For inference only the input image is shown and the system generates reconstructed human image as well as its appropriate segmentation and pose

MultiPathCNN to Construct Target Segmentation Architecture and Training



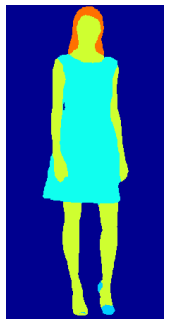
- Auto-encoder system with small latent space vector $\underline{z}$
- Parallel decoder paths (Image, Clothes Segmentation, Pose image) share weights on first layers
- Learns to synchronize during training
- For inference only the input image is shown and the system generates reconstructed human image as well as its appropriate segmentation and pose

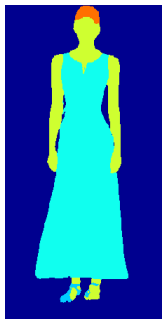
Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

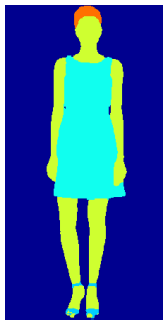
User

Dress

Pose





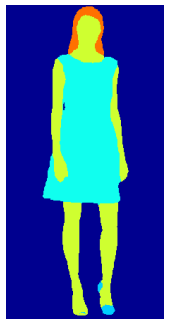


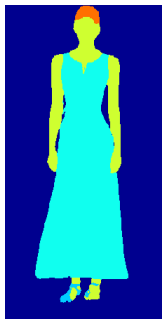
Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

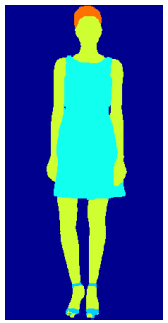
User

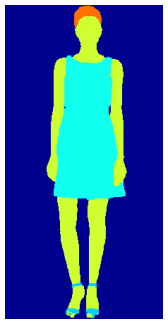
Dress

Pose







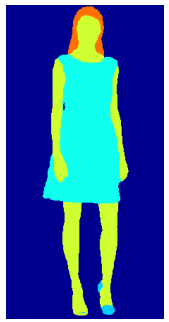
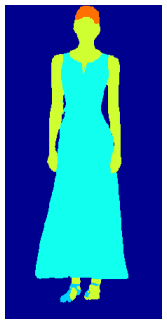
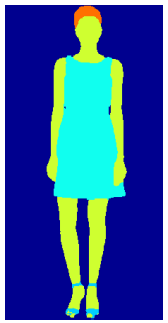


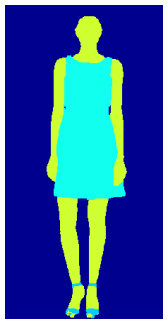
1. Start with initial pose segmentation

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

User Dress Pose



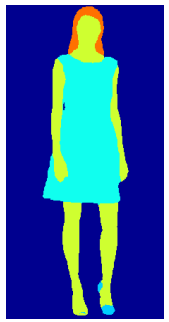
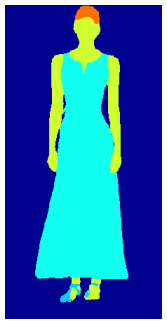
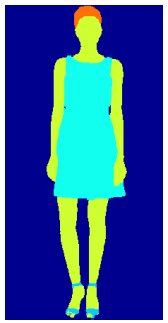


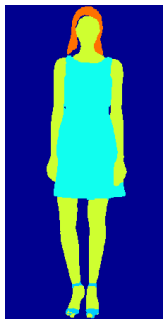
1. Start with initial pose segmentation
2. Remove hair

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

User Dress Pose



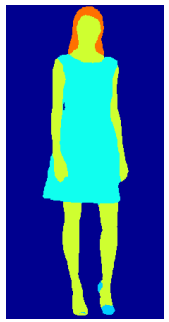
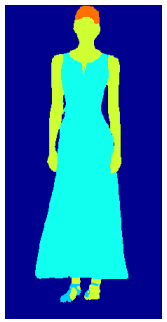
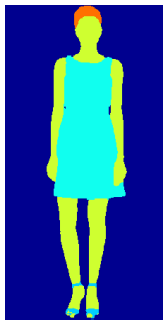


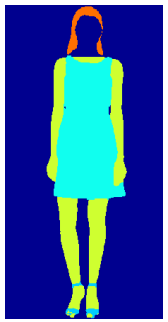
1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

User Dress Pose



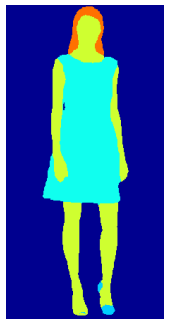
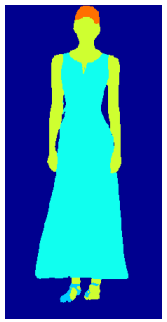
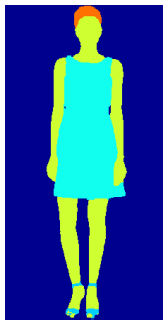


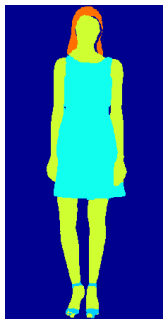
1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user
4. Remove face

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

User Dress Pose





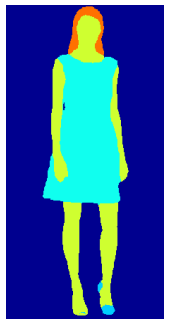
1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user
4. Remove face
5. Add face from user

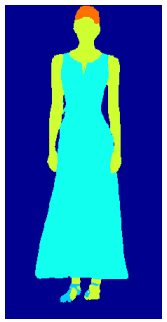
Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

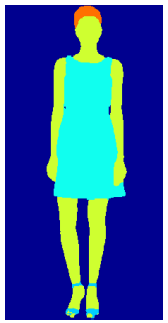
User

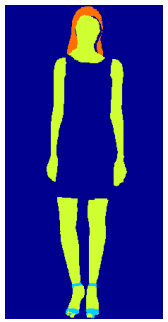
Dress

Pose







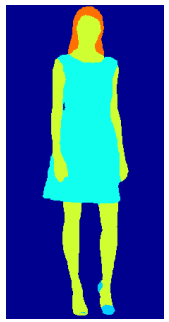
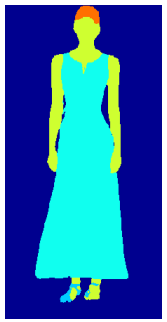
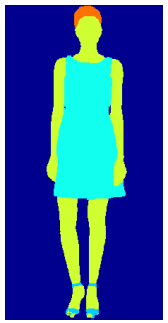


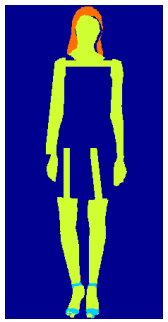
1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user
4. Remove face
5. Add face from user
6. Remove dress

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

User Dress Pose



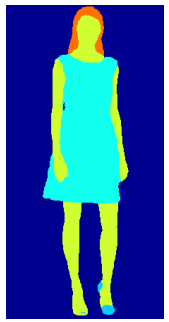
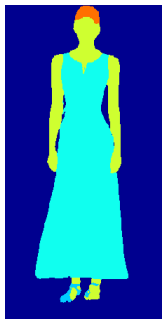
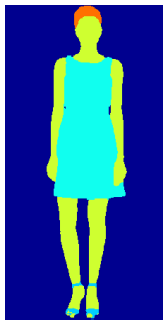


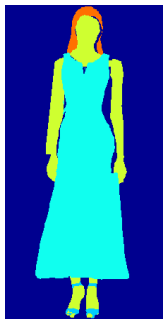
1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user
4. Remove face
5. Add face from user
6. Remove dress
7. Add pose lines

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

User Dress Pose



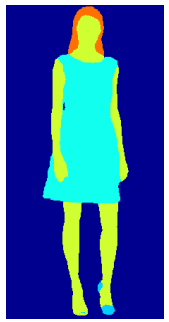
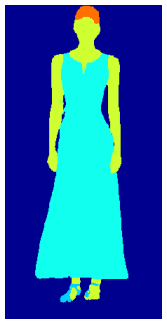
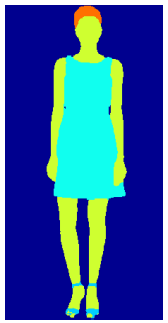


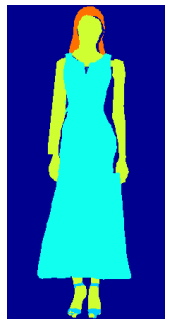
1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user
4. Remove face
5. Add face from user
6. Remove dress
7. Add pose lines
8. Add new dress

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

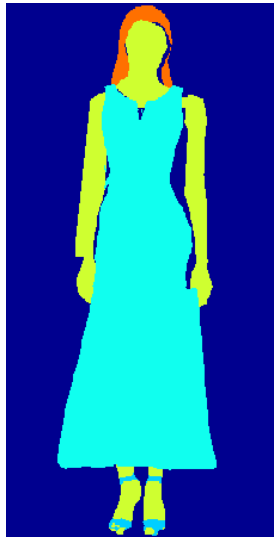
User Dress Pose





1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user
4. Remove face
5. Add face from user
6. Remove dress
7. Add pose lines
8. Add new dress

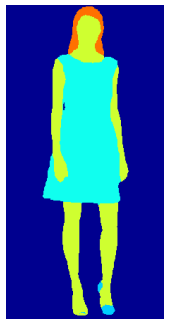
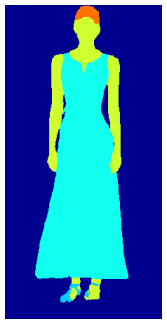
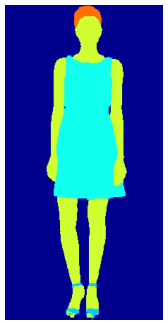


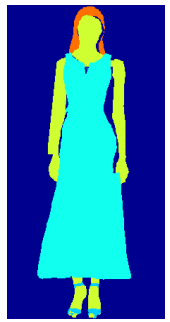
$\hat{S}_{Out}^{Coarse}$ – not consistent!

Creation of the Coarse Target Segmentation - $\hat{S}_{Coarse}$

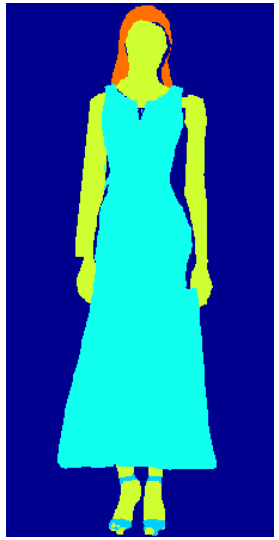
User Dress Pose





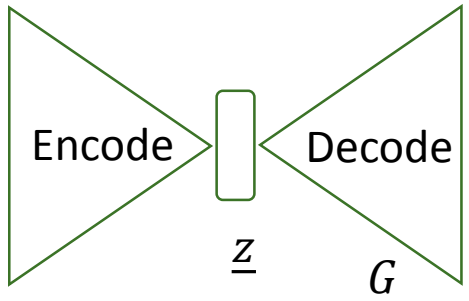
1. Start with initial pose segmentation
2. Remove hair
3. Add hair from user
4. Remove face
5. Add face from user
6. Remove dress
7. Add pose lines
8. Add new dress



$\hat{S}_{Out}^{Coarse}$ – not consistent!

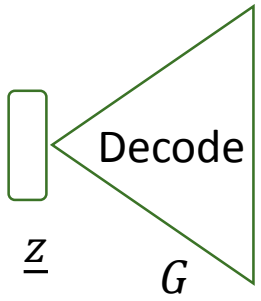
Use MultiPathCNN to Fix It

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*



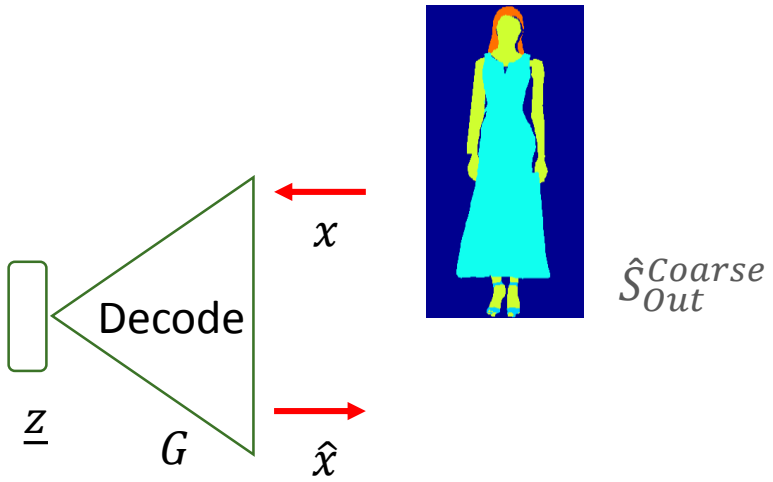
- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*



- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*



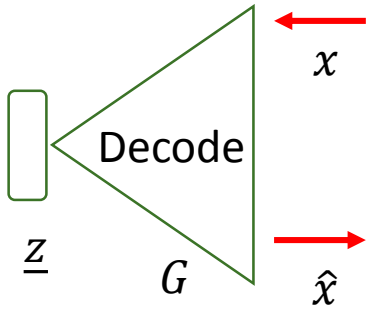
$\hat{S}_{Out}^{Coarse}$

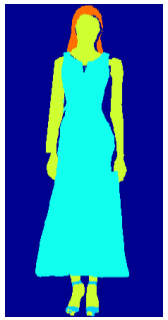
Apply backward procedure

a. Start with initial pose segmentation

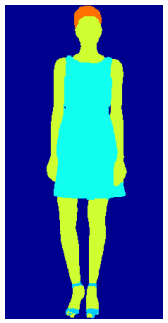
- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$



$\hat{S}_{Out}$

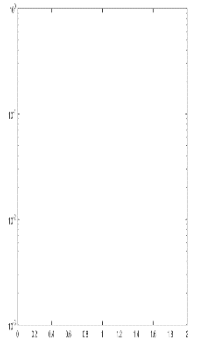
$\boldsymbol{Iteration_0}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

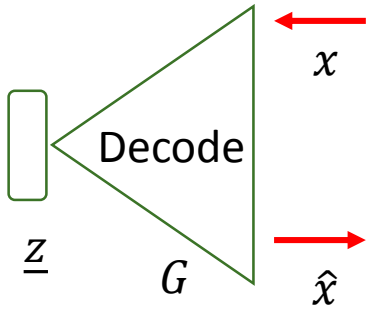
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

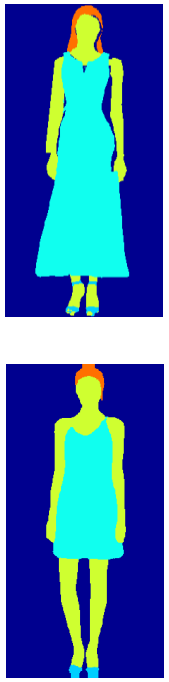


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

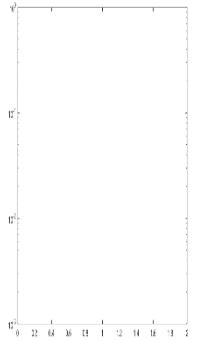
$\boldsymbol{Iteration_1}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

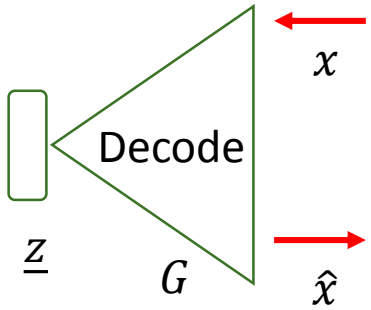
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

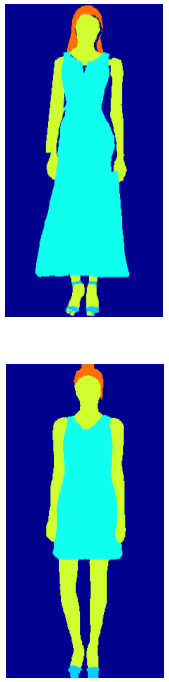


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

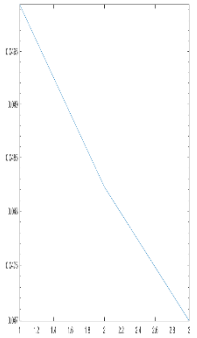
$\boldsymbol{Iteration_2}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

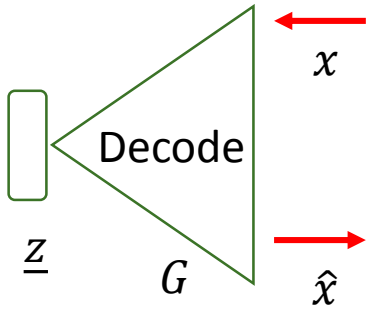
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

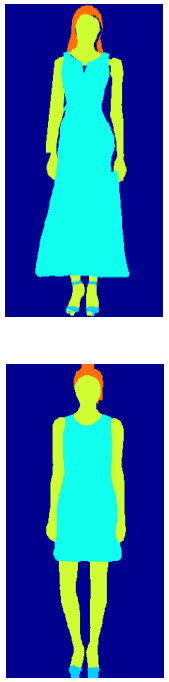


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

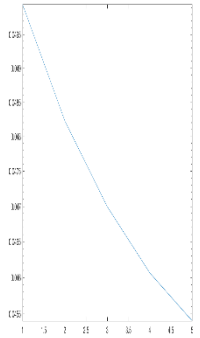
$\boldsymbol{Iteration_3}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

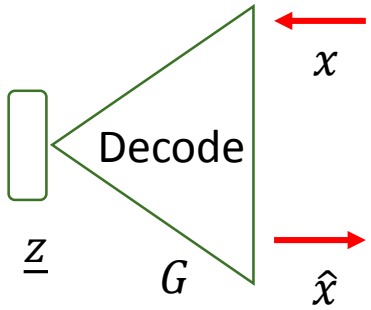
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

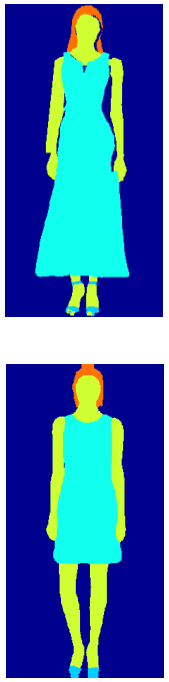


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

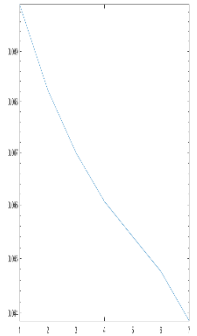
$\boldsymbol{Iteration_4}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

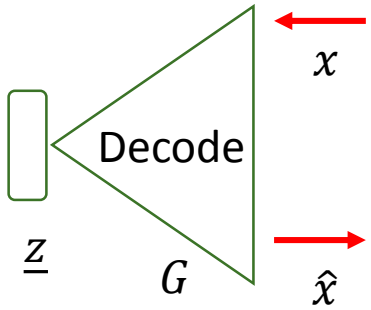
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

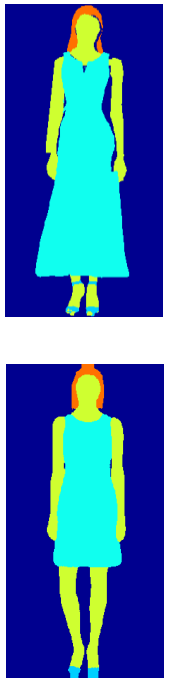


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

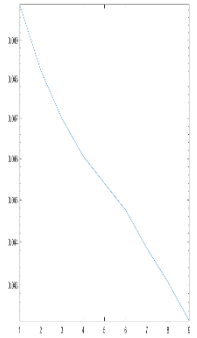
$\boldsymbol{Iteration_5}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

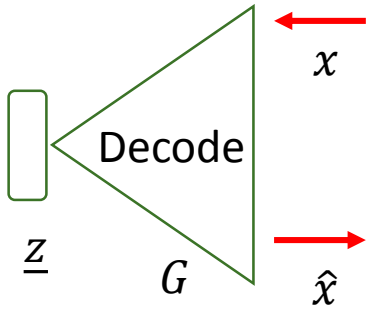
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

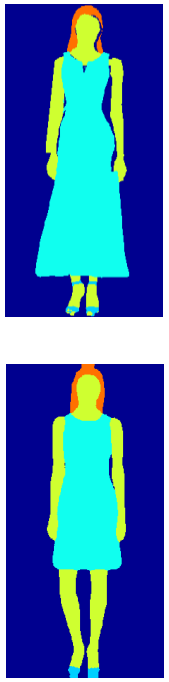


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

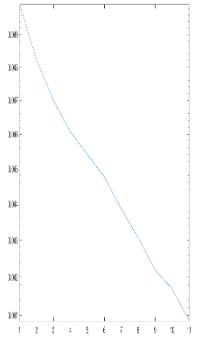
$\boldsymbol{Iteration_6}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

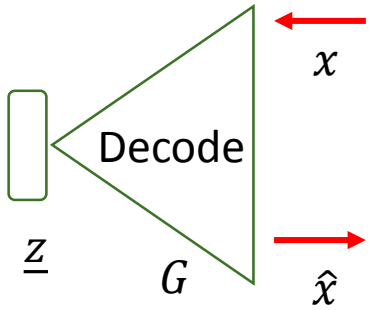
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

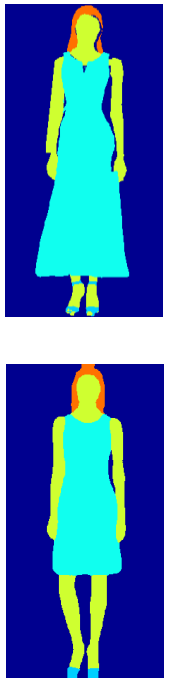


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

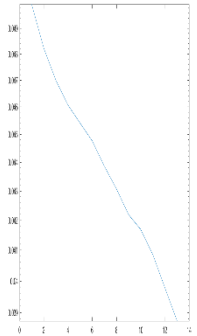
$\boldsymbol{Iteration_7}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

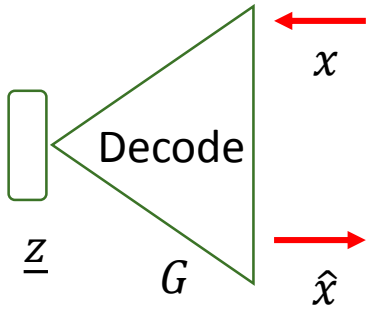
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

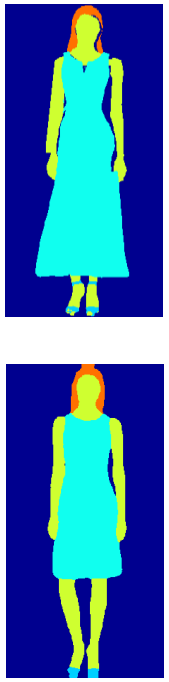


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

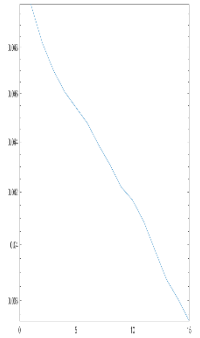
$\boldsymbol{Iteration_8}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

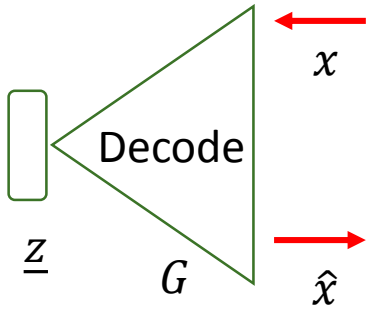
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

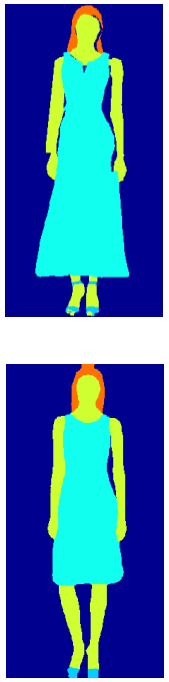


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

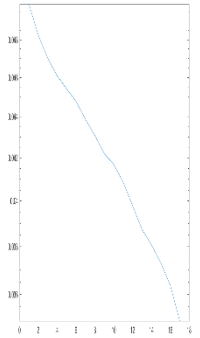
$\boldsymbol{Iteration_9}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

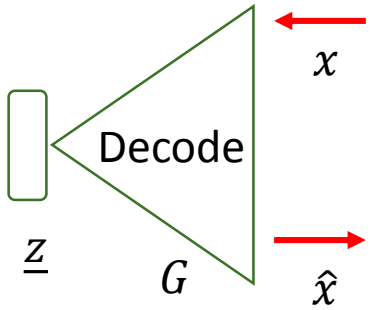
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

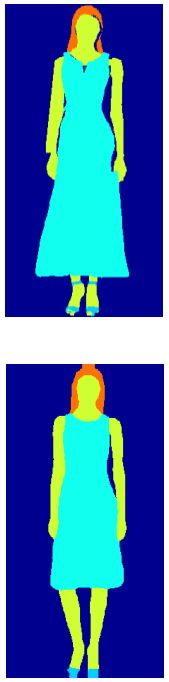


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

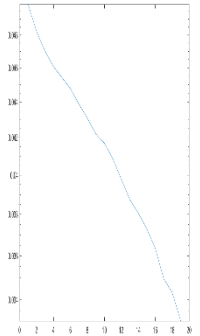
$\boldsymbol{Iteration_{10}}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

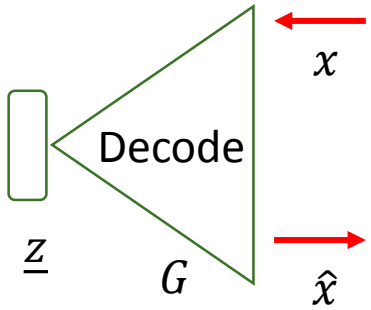
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

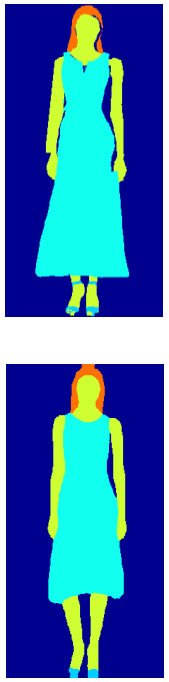


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

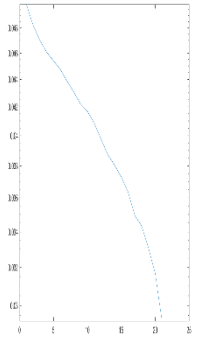
$\boldsymbol{Iteration_{11}}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

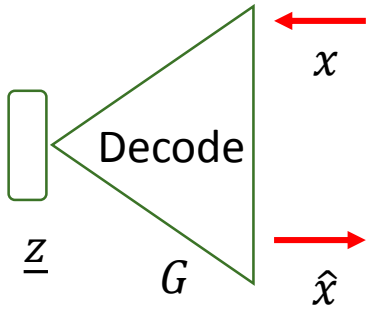
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

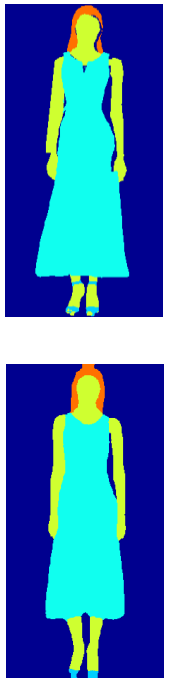


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

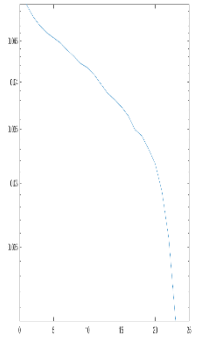
$\boldsymbol{Iteration_{12}}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

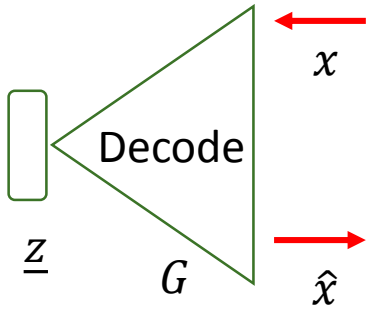
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

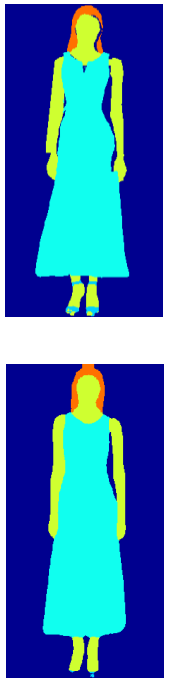


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

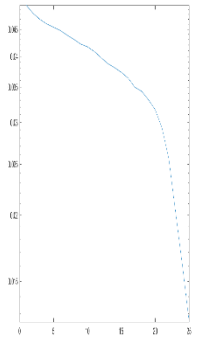
$\boldsymbol{Iteration_{13}}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

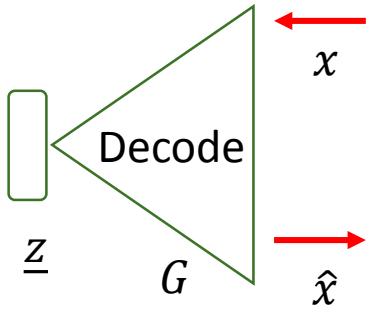
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

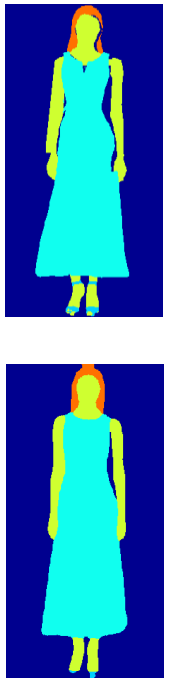


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

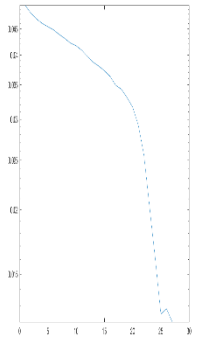
$\boldsymbol{Iteration_{14}}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

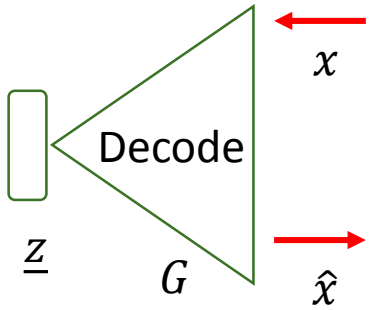
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

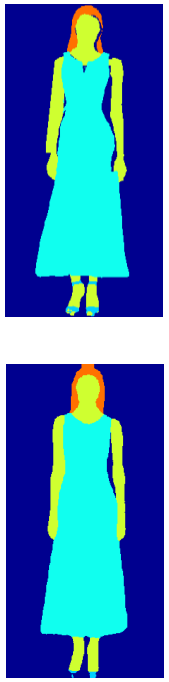


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

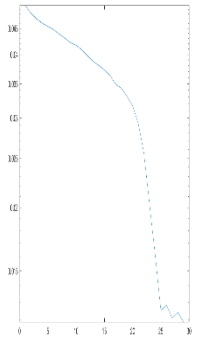
$\boldsymbol{Iteration_{15}}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

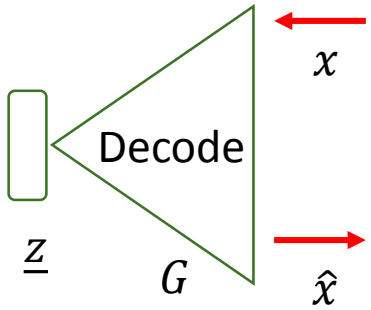
$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$

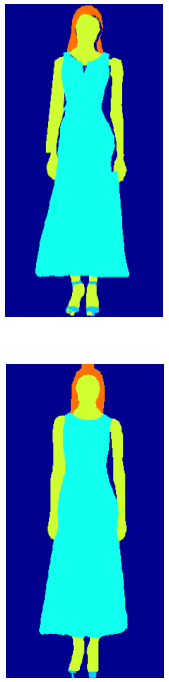


Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

Creation of the Final Segmentation - $\hat{S}_{OUT}$ by *Backward Procedure*





$\hat{S}_{Out}^{Coarse}$

$\hat{S}_{Out}$

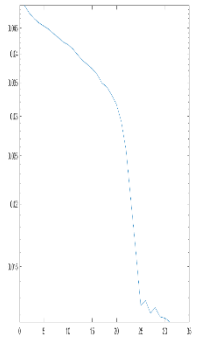
$\boldsymbol{Iteration_{16}}$

Apply backward procedure

a. Start with initial pose segmentation

b. Iterate until convergence

$$Err = \left|\hat{S}_{Out}^{Coarse} - \hat{S}_{Out}\right|$$



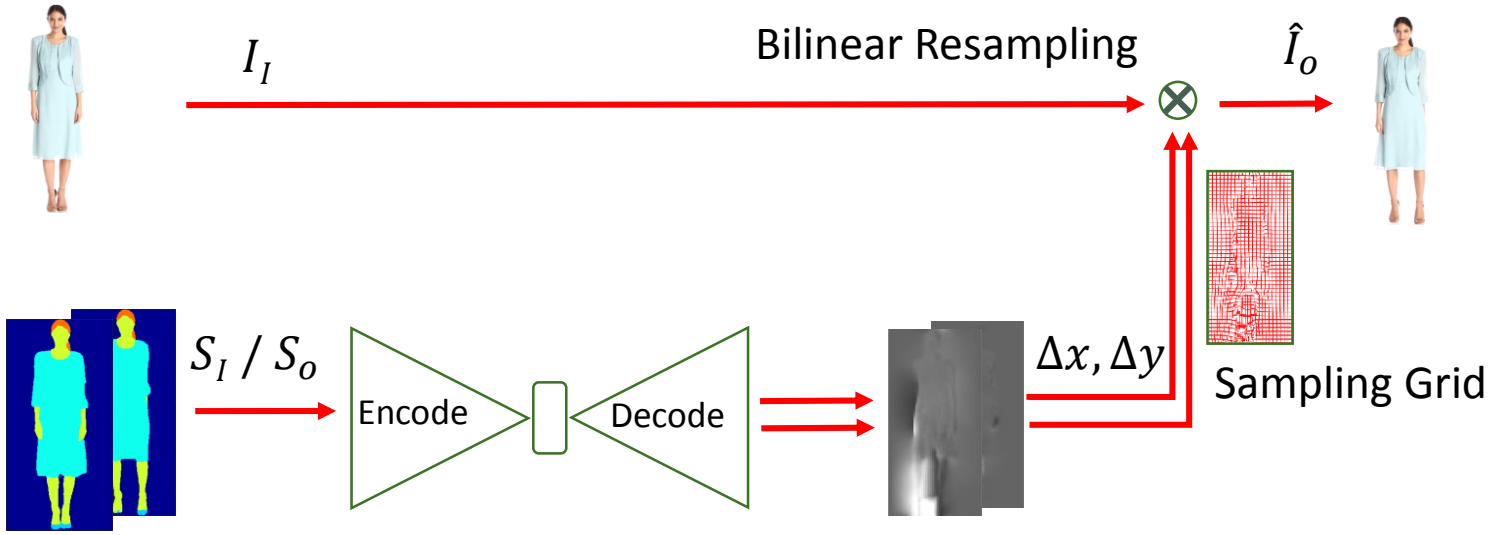
Err

- x – Image that needs to be reconstructed
- Present x ($\hat{S}_{Out}^{Coarse}$) at the output of the decoder (generator)
- Back propagate through the G to find the best latent vector $\underline{z}$ that compensates for the changes in x.
- Use the forward pass through the decoder to get a reconstruction $\hat{x}$ ($\hat{S}_{Out}$).

ReMapCNN – Optical Flow with Local Transform to Preserve High-Resolution Details

Motivation:

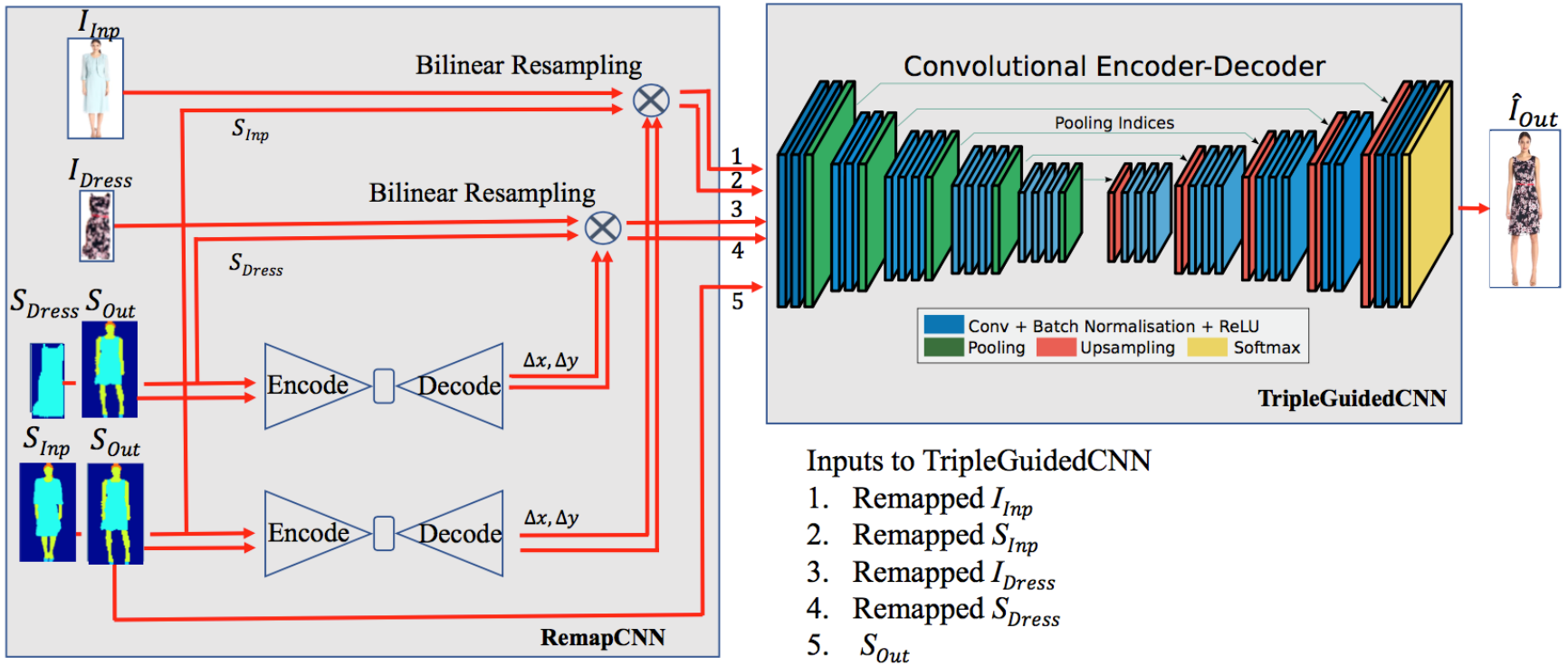
- CNN cannot recreate details when the geometry of the image changes
- GAN ‘invents’ details
- Optical flow ‘moves’ pixels rather than ‘invents’ them
- Used as pre- or post- processing step to TripleGuidedCNN



Loss: $\varepsilon = \left|I_o - \hat{I}_o\right|_{L_1} + \alpha(|\nabla x| + |\nabla y|) * M_{Not\ Boundaries}$

(Total Variation as a Regularization term)

Combined ReMapCNN and TripleGuidedCNN



Inputs to TripleGuidedCNN

1. Remapped I_{Inp}
2. Remapped S_{Inp}
3. Remapped I_{Dress}
4. Remapped S_{Dress}
5. S_{Out}

- Combining a ReMapCNN as a pre-processing to TripleGuidedCNN enables end-to-end solution producing qualitive final result
  - ReMapCNN – achieves high res details transfer
  - TripleGuidedCNN – completes missing details

Some examples

- Dress transfer to the same pose

- Dress transfer to the different pose

Sleeves revealed

- Dress transfer to the different pose

- Swapping user and the donor

Sample Results and User Study

Sample Results and User Study

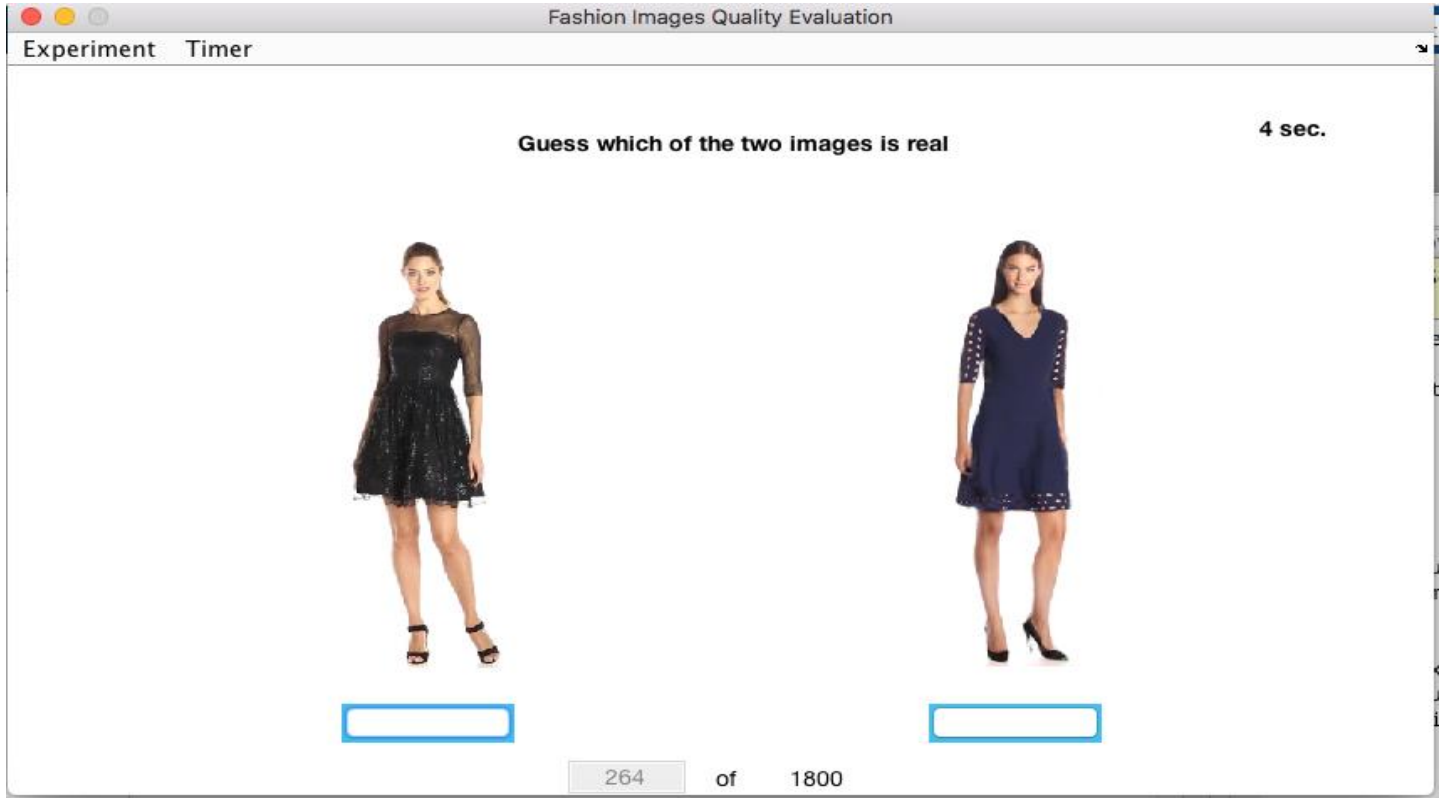
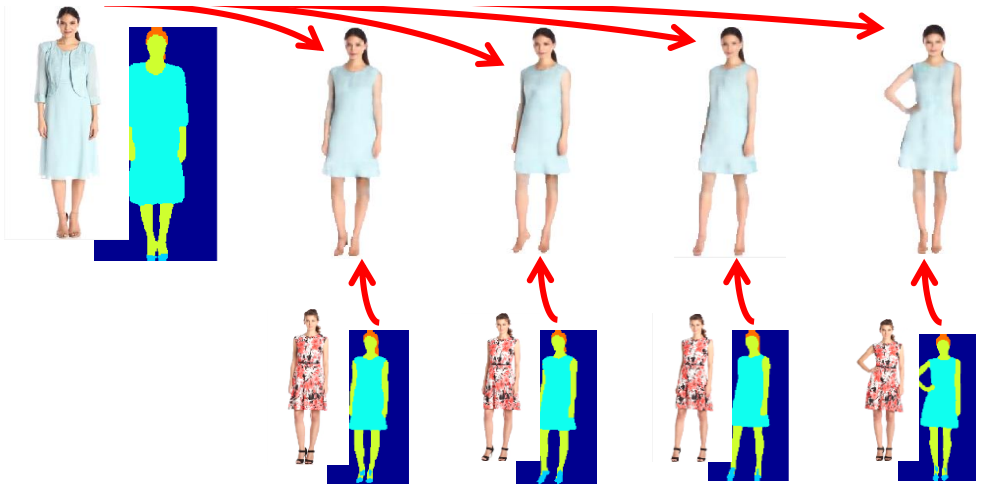


Table 1: User Study on Pose-and-Dress set. Average "fooling" rate.

Perceptual time per example	1 sec.	2 sec.	5 sec.
No ReMap	17.6%	5.6%	0%
ReMap as post-processing (VITON-like)[2]	48.0%	43.8%	29.7%
ReMap as pre-processing	47.6%	44.1%	30.4%
ReMap and TripleGuidedCNN trained jointly	**48.1%**	**46.1%**	**34.2%**

Generating Videos from Single Image

User Image | Catalog Dress | Pose Constraint | Result Image



Frames extracted from template video

Final Video Clip

Final Video Clip
Temporarily Filtered

Conclusions

- Transfer new dress item to user while changing pose
- Pose-and-Dress dataset
- Backward operating MultiPathCNN system to create target consistent segmentation
- TripleGuidedCNN to make a low-res dress transfer
- RemapCNN to add high-res details
- Creating video clip showing user in a new dress

- General technology for the High Resolution Geometrical Cross-Domain Transfer

Thank You!

Ilia Vitsnudel

Liza Potikha