DressUp: A virtual Try-on applications for Apparels using Edge computing

Aylmer Britto Rex Harison(arexhari) Mentor:Shilpa George

Instructors: Dr. Mahadev Satyanarayan,Dr. Asim Smailagic

Objective

- Closure of Trial rooms at retail stores during covid, Longer wait time for trial rooms at retail stores and limitation on the number of items is making shopping inconvenient.
- We propose an AR system that lets users to virtually try on clothes without stitching the head to the given mannequin
- Superimpose the cloth on the given user:
  - estimate the pose of the user
  - warps the clothing to generate a photorealistic fitting resul

Background

PF-AFN[1]

- Parser Free Appearance Flow Network (PF-AFN) - CVPR'21
- Warp Model (AFWM) to establish accurate dense correspondences between the person image and the clothes image
- Generative Model (GM) synthesize output image

Mediapipe

- Mediapipe is a cross platform library that offers customizable ML solutions for streaming media
- We use it to segment the person from a noisy background

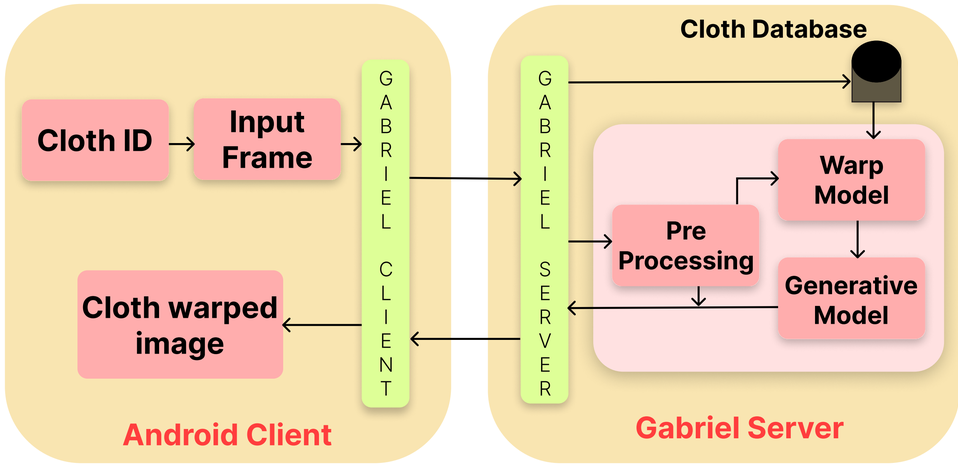
Gabriel cloudlet server[2]

- Gabriel is a framework for cognitive edge applications using cloudlets
- Assists with tight end-to-end latency bound on compute intensive operations

Need for Edge Computing

- PF-AFN: Our primary algorithm for warping the clothes demands high computaion resources as it is comprised of two deep neural networks inferencing
- Offloading it to a cloudlet compared to a cloud in our case would save transmission costs.

Block diagram and algorithm



1. The Cloth Id and Input frames are captured and passed to the gabriel client layer to pack the protobuf payload
2. The Gabriel server i.e., the top layer at the cloudlet, passes the input frame to the dressUp engine as a numpy array
3. The dressup Engine does appropriate preprocessing with mediapipe to check if humans are present in the frame, removes the external noise in the background and passes it to the warp model. If no humans are present, then the input frame is just sent back to the client without any warping.
4. The warp model marks accurate dense correspondences between the person image and the clothes image.
5. Then it passes the image to the generative model to sythesizes the output image.
6. The gabriel client parses the image and displays it to the user

Static Image Results

DressUp Engine Performance

DressUp Engine Start Time = 0.7s

0.03s	0.03s	0.09s
Mediapipe Preprocessing	Warp Model Inferencing	Generative Model Inferencing

6 FPS

Overhead observed due to network and payload packing for input frames = 0.5 FPS

Challenges Faced

- To find the apt Virtual Try On algorithm that fits our functional and system requirements
- To host mediapipe, PF-AFN and gabriel server in the same environment
- To make an object oriented modular inference script for PF-AFN to align with gabriel server and preprocessing pipeline(mediapie)
- The video performance was poor and unstable when exposed to enviroments with enormous noise

Course Takeaways

- The *cyber-foraging* techniques discussed in the reading materials helped me to figure out the right routies to off-load heavy computaions in the cloudlet
- *Ubiquitous data access* lectures inspired to shape the architechture with respect to introduce clothes database instead of appending the cloth and edge images to the protobuf payload from android client

Future works

- To pose DressUp Engine as a portable Virtual machine being to able offload and start serving on demand at any cloudlet.

References

[1] Ge, Y., Song, Y., Zhang, R., Ge, C., Liu, W. and Luo, P., 202 Parser-free virtual try-on via distilling appearance flows.
[2] Satyanarayanan, M., 2013, June. Cloudlets: At the leading edge of cloud-mobile convergence.