Final Project Report
Earring Tryon

Jessie Wang
COMS-BC3168 Deep Learning for Computer Graphics
Dr. Corey Toler-Franklin

May 12, 2026

Abstract

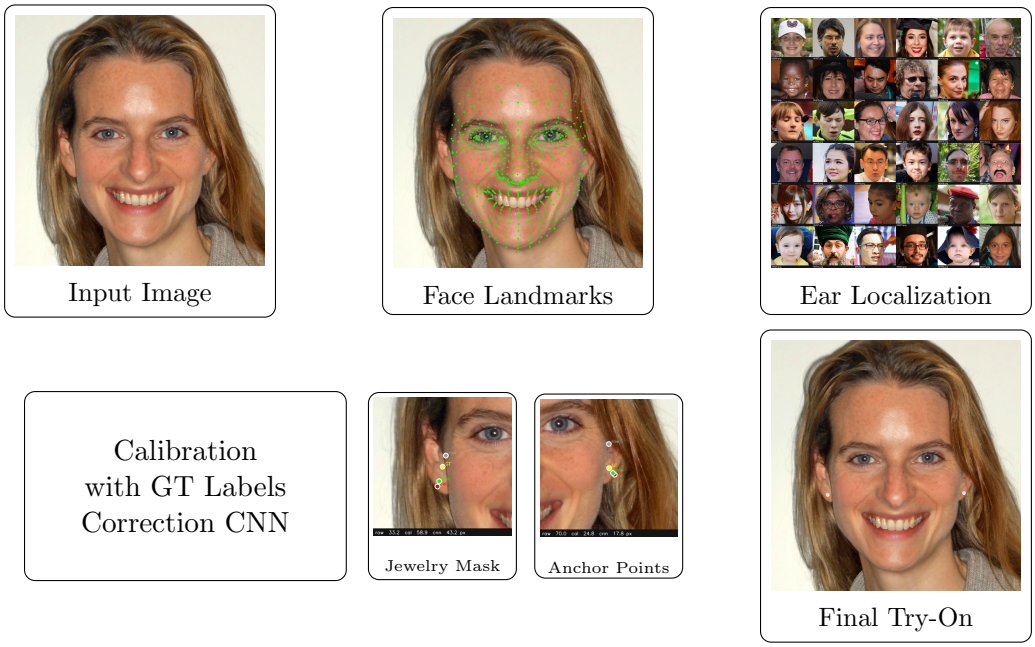

Figure 1: Pipeline for the virtual jewelry try-on system.

Introduction

Image based virtual try on has become increasingly popular through usage with online shopping, especially with the emergence of AI fashion as a profitable industry. Our project specifically focuses on the challenge of jewelry try on, which has been neglected by most previous models like VITON that focused on the cloth warping aspect, and the specific fit of clothing in certain styles. OmniTry on the other hand, is a model that attempts to fill this gap by producing a model that allows for object try-ons, ranging from shoes to jewelry. Though they employ several other things like MLLM, masked full attention and two stage training, their novel contribution is rooted in the traceless

erasing process employed to remove objects with no trace in order to produce paired training data efficiently. This reflects one key obstacle in these problems - assembling the appropriate dataset with ground truth labels that efficiently train the model.

For our project, this posed to be a large problem, as there were not many datasets that had annotated data of detailed parts of the ear. Ultimately we resorted to hand annotated data as the ground truth placements for where the jewelry was meant to go on the ear. We use them to calibrate the initial landmarks produced by MediaPipe, and then implement a correction CNN to produce final anchor points for the earrings.

There are several ways in which we could improve upon this implementation. For the identification of the ear lobe, we attempted using color thresholding for metallic pixels to find where the earring was relative to the ear in images of models wearing earrings. This was performing well so we ended up going with the hand annotations, however we could attempt to use a U-net instead of thresholding to avoid picking up on other metallic things that are not the jewelry itself.

Previous Work

Most previous work on virtual tryon has focused on clothing - with significant milestones being achieved by models like VITON and HR-VTON. In the survey Song et al, 2023, they categorize current methods into physical based simulation, real acquisition and image generation. OmniTry, presented in Tang et al, 2025, identifies this gap in previous works. While VTON has explored shoe tryon, there lacks a framework on wearable objects, which OmniTry tackles.

Some common implementations of recent models include using diffusion models, GAN, or incorporating some version of a large language model. The first generation of models like CAGAN (2017) and VITON (2018) focused on warping the clothing correctly around the pose and body structure of the person.

A challenge which emerged was the burden of extracting training data. Ideally one would need an image of a person wearing some clothing A, a product image of clothing B and then the ground truth image of the person wearing clothing B if one is training the network to learn how to put clothing B on a person who was originally wearing clothing A. Clothing agnostic approach removes what the person was originally wearing and uses this clothing agnostic model to pair with any product image of clothing B and then compare the loss to the ground truth image of a person wearing clothing B - only requiring two images of the same model as opposed to two images in different clothing.

Clothing warping strategies have evolved from things like Thin Plate Spline (TPS) to multistage pipelines to GAN based methods that use U-nets and StyleGAN - which tries to learn different styles of clothing to better fit their correct forms. Most recent models however have shifted to using diffusion models. By taking advantage of the forward process that adds noise and the reverse process that reconstructs the image from noise, one can guide the process with conditions at each noise removal step.

In terms of OmniTry, they attempt to use mask free localization - identifying where an item should go on a person without explicit instructions. They achieve this by using a large language

model (Qwen-VL) to list all wearable items in an image, then detecting and segmenting the items to remove the item - creating pairs of data. They also use masked full attention to concatenate tokens from the individual images of the person and sunglasses so the two streams do not corrupt each other. Other than OmniTry, there does not seem to be other try on models that tackle the specific task of object try on.

Data

The data used was from this kaggle dataset https://www.kaggle.com/datasets/arnaud58/flickrfaceshq-dataset-ffhq. It consists of 512x512 resolution images of faces. Only one folder of the data: 06000, was used in the training and testing of this project. The data provided clear images of faces, though there was no specific constraint on both ears being available thus a lot of the images were not used in the manual annotation process.

Technical Description

MediaPipe was used as a prior to identify features close to the ear, however since the Face Landmarker (Kartynnik et al., 2019) fails to produce specific features on the ear - lobe, cartilage or helix, this could only be used as a first pass for the data. We obtained 478 facial landmarks with the Face Landmarker, and selected indices 234 and 454 as the first version of the anchor, which correspond to the right and left ear cartilage.

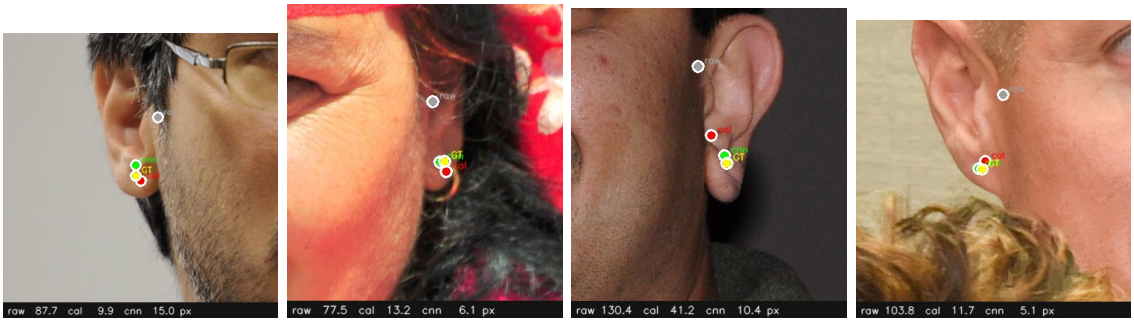

Figure 2: Anchors produced from raw MediaPipe landmarks (grey), Ground truth placements (yellow), MediaPipe landmarks calibrated with ground truth placements (red), and CNN predicted (green).

We attempted to explore other options at this point such as using a HSV threshold to mask out the metallic bits of the earring on a separate dataset (https://huggingface.co/datasets/raresense/jewelry-dwpose-dataset) containing images of people wearing earrings. This method often succeeded in terms of masking the correct parts for the earring - when the data was conveniently cropped where the earring was likely the main metallic object in frame, however depending on earring size and head orientation, getting from knowing where the earring was to identifying the precise anchor point remained challenging. Another possible method we looked into was exploring existing labeled datasets of images of ears such as AWE (Annotated Web Ears). Though these datasets contained a variety of subjects, the intended purpose for them seemed to be ear recognition in

unconstrained environments as the annotations mainly consisted of head orientation and the only labeled physical feature was the tragus - which was not the feature we were attempting to place the earring on.

Ultimately we hand labeled ear lobe anchor points to 500 images, from which there were 337 labeled images, others were skipped if ears were not visible or covered, and 396 individually labeled ears (some had both sides visible some had only one). The pixels as the left and right anchors were then used as the ground truth labels to calibrate the anchors produced by MediaPipe.

The calibration process was taking the mean of the offset of the MediaPipe anchors from the ground truth in pixels for the left and right sides individually. $\Delta\bar{x} = \frac{1}{N}\sum_{i=1}^{N}(p_{gt,x} - p_{mp,x}), \Delta\bar{y} = \frac{1}{N}\sum_{i=1}^{N}(p_{gt,y} - p_{mp,y})$ Instead of learning these two parameters with the CNN, we use the mean as it is a constant shift that does not necessitate a CNN.

For the CNN itself, we use a 96 x 96 RGB patch that is centered on the calibrated anchor as the input, and produce offsets in both dimensions (x,y), which are patch normalized. The network consists of four conv blocks with channels growing from 3 at input to 128 at the fourth conv block, while the spatial size shrinks from 96x96 to 6x6 using max pooling. This pattern of doubling the amount of channels as the amount of space is halved established in VGG (Simonyan & Zisserman, 2015) allows for the network to work well for this problem that requires a precise identification of a feature. The features are then flattened to a 4608 dimension vector that is compressed with FC layers to the 2 dimensional output that represents the correction offset.

Training is done with an Adam optimizer with a learning rate of 1^{-3}, over 40 epochs and batch size 32, early stopping while waiting 8 epochs for improvement on validation loss. The CNN is trained on the left side patches while the right side images are mirrored horizontally and the prediction is mirrored back, as a method to double the training data. The randomness across Numpy, PyTorch and the split among training validation and test sets are seeded with seed=42.

Stage	Layer type	Spatial size	Channel in	Channel out
Input	—	96×96	—	3
Conv-1	Conv + ReLU + MaxPool	48×48	3	16
Conv-2	Conv + ReLU + MaxPool	24×24	16	32
Conv-3	Conv + ReLU + MaxPool	12×12	32	64
Conv-4	Conv + ReLU + MaxPool	6×6	64	128
Flatten	—	—	—	4608
FC-1	Linear + ReLU + Dropout(0.2)	—	4608	256
FC-2	Linear + ReLU	—	256	64
Output	Linear	—	64	2

Table 1: CorrectionCNN architecture. All convolutions use 3×3 kernels with padding 1; pooling uses 2×2 max-pooling with stride 2. The output is a 2-dimensional residual offset $(\delta x, \delta y)$.

Results and Analysis

Figure 3: Sample results of the try on

This figure below show the error between the pixel difference between the ground truth label of anchor and the anchor identified in each stage. The error with just raw MediaPipe labels can be seen to be very large while for CNN there still remains outliers but the error is reduced significantly.

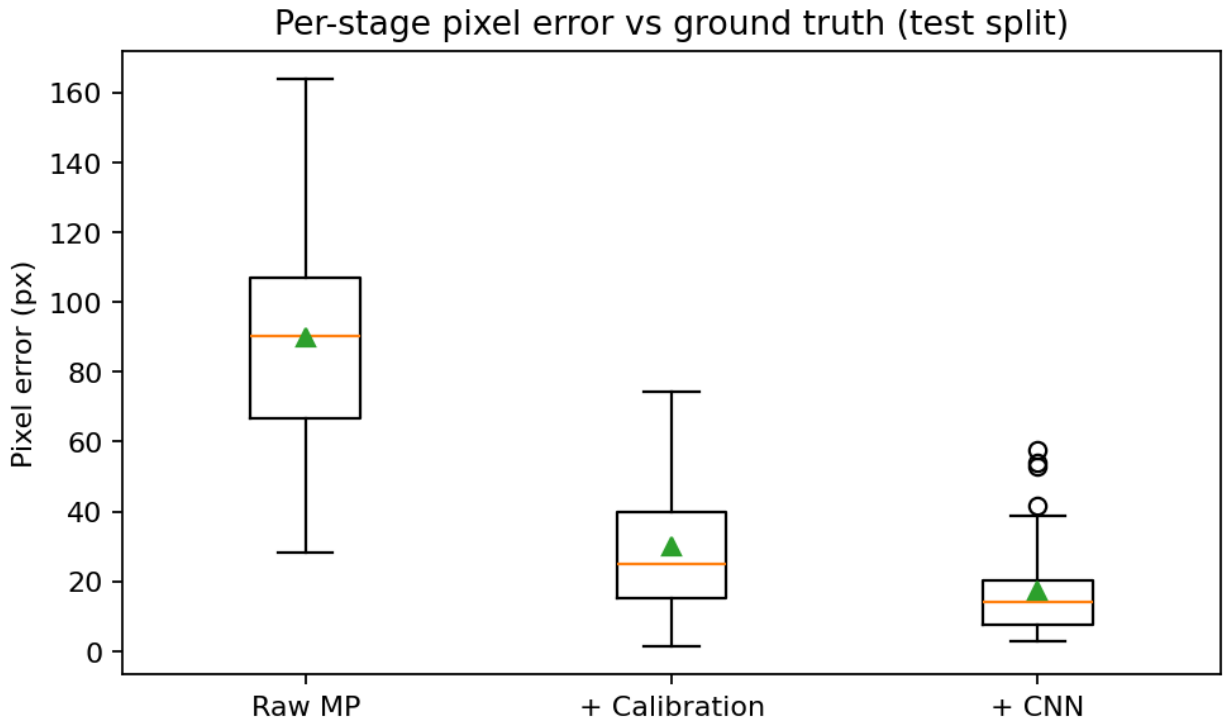

Figure 4: Box plot of pixel error for each stage of the implementation

Discussion and Improvements

One possible improvement mentioned previously is to use U-nets or other deep learning techniques to implement the locating of earrings in the dataset which included images of models wearing jewelry, instead of an HSV threshold. Other improvements include recognizing when there is only one side or neither sides with ears available, and not attaching an earring in that case, as currently, as seen in the results, there are several cases where the earring is automatically put onto the image when the ear is actually not visible and it forces a fit.

Citations

1. Song, D., Zhang, X., Zhou, J., Nie, W., Tong, R., Kankanhalli, M., Liu, A. A. (2024). Image-Based Virtual Try-On: A Survey. arXiv preprint arXiv:2311.04811v4. https://arxiv.org/abs/2311.04811

2. Feng, Y., Zhang, L., Cao, H., Chen, Y., Feng, X., Cao, J., Wu, Y., Wang, B. (2025). Omni-Try: Virtual Try-On Anything without Masks. arXiv preprint arXiv:2508.13632v1. https://arxiv.org/abs/2508.136

3. Zhu, L., Yang, D., Zhu, T., et al. (2023). TryOnDiffusion: A tale of two unets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4606-4615.

4. Han, X., Wu, Z., Wu, Z., et al. (2018). VITON: An Image-based Virtual Try-on Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7543-7552. https://arxiv.org/abs/1711.10972

5. Kartynnik, Y., Ablavatski, A., Grishchenko, I., Grundmann, M. (2019). Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

6. Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In International Conference on Learning Representations (ICLR). arXiv preprint arXiv:1409.1556

Datasets used and referenced:
1. FFHQ Dataset (Flickr Faces HQ): https://www.kaggle.com/datasets/arnaud58/flickrfaceshq-dataset-ffhq

2. Jewelry-DWPose Dataset: https://huggingface.co/datasets/raresense/jewelry-dwpose-dataset
3. AWE (Annotated Web Ears) Dataset: