**TryOnDiffusion: A Tale of Two UNets (Architecture)**

**1. Base Diffusion Model:**

- Architecture: Parallel-UNet

- Input Resolution: 128×128

- Layers:

- Encoder-decoder architecture with skip connections

- Multiple layers of convolutions in encoder and decoder parts

- Possibly normalization layers (e.g., BatchNormalization)

- Activation functions (e.g., ReLU) after each convolution

- Output layer to predict the noise that corrupts the input image

**2. Super-Resolution (SR) Diffusion Models:**

- Two SR diffusion models, one for 128×128→256×256 and another for 256×256→1024×1024 resolutions.

- Architecture: Parallel-UNet for 128×128→256×256 and Efficient-UNet for 256×256→1024×1024

- Layers:

- Similar architecture to the base diffusion model but with adjustments for the target resolution

- Increased number of filters and layers in the decoder part for upsampling

- Additional layers for handling noise conditioning augmentation

- Possibly residual connections for better gradient flow

- No attention layers for the 1024×1024 SR diffusion model

**3. Parallel-UNet:**

- Architecture: Parallel-UNet

- Input Resolution: 128×128

- Layers:

- Two UNets for handling the person and garment, respectively

- Both UNets should consist of encoder-decoder architecture

- Additional attention mechanism for implicit warping between garment and person streams

- Modulation of features using FiLM across all scales

- Pose embeddings used for guiding warp and blend process

- Possibly residual connections within UNets for better information flow

**4. Training Details:**

- Loss functions (e.g., weighted denoising score matching objective)

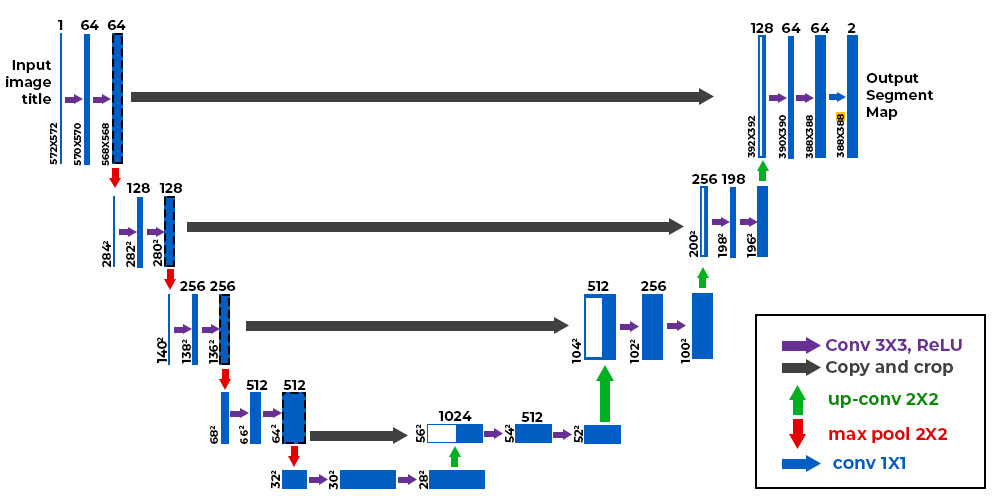
- Optimization algorithm (e.g., Adam optimizer)

- Learning rate schedule

- Data augmentation techniques (e.g., random Gaussian noise augmentation)

- Evaluation metrics (e.g., PSNR, SSIM)

**U – Network:**



<https://www.geeksforgeeks.org/u-net-architecture-explained/>

**Attention Layers:**

Query, Key, Value: In attention mechanisms, the input is typically split into three parts: query, key, and value. These parts can be derived from the same input or computed separately.

Similarity: The attention mechanism computes the similarity between the query and the keys, usually using dot product, cosine similarity, or other similarity measures.

Weights: Based on the computed similarities, attention weights are generated to determine how much focus to place on each element of the value.

**Peak Signal-to-Noise Ratio (PSNR):**

PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

It's calculated as the ratio of the maximum possible power of a signal (usually the maximum possible pixel value) to the power of the noise that corrupts the fidelity of the signal.

PSNR is expressed in decibels (dB) and is calculated using the mean squared error (MSE) between the original and distorted images.

PSNR provides a measure of how much the quality of an image deteriorates when it is subject to noise or distortion. Higher PSNR values indicate lower levels of distortion and higher image quality.

**Structural Similarity Index (SSIM):**

SSIM is a metric that measures the structural similarity between two images.

It considers the luminance, contrast, and structure of the images to provide a more comprehensive measure of similarity compared to PSNR.

SSIM compares local patterns of pixel intensities rather than just the pixel values themselves.

SSIM ranges between -1 and 1, where a value of 1 indicates perfect similarity between the images.

SSIM is calculated by comparing the luminance, contrast, and structure similarity between the original and distorted images.

**Virtual Try-On with Pose-Garment Keypoints Guided Inpainting (Architecture)**

***Stage 1: Pose-Oriented Garment Keypoints Prediction***

**Graph Neural Network Architecture:**

- Two-Stream Graph Neural Network:

- Main Stream:

- Input: Garment graph (32 nodes representing garment keypoints).

- Graph Convolution Blocks: Extract features from garment keypoints.

- Side Stream:

- Input: Pose graph (10 nodes representing upper human body joints, 18 directional

edges representing human skeletons).

- Embedding: Hierarchically provides pose information to the main stream at different feature levels.

- Graph Convolution Blocks:

- Used for feature extraction from both garment and pose graphs.

***Stage 2: Segmentation Map Generation, Cloth Warping, and Person Image Recomposition***

**- Target Segmentation Map Generation:**

- Autoencoder Architecture:

- Input: Human skeleton and garment contour sketches, garment-agnostic segmentation map.

- Output: Target segmentation map.

- Architecture: Encoder-decoder structure with convolutional layers for feature extraction and upsampling layers for segmentation map generation.

**- Cloth Warping:**

- Warping Architecture:

- Utilizes paired original and pose-oriented keypoints to warp each sub-segment of the garment individually.

- Sub-segments: Left low, left up, center, right up, and right low.

- Person Image Recomposition:

- Recomposition Architecture:

- Combines warped garment images with the target segmentation map to recompose the person image.

***Stage 3: Semantic-conditioned Inpainting***

**- Inpainting Process:**

- Denoising Diffusion Model Architecture:

- Input: Recomposed person image with missing regions.

- Conditioning: Segmentation map and known pixels.

- Diffusion Process: Progressively inpaints missing regions conditioned on the segmentation map and known pixels.

- Architecture: Utilizes a denoising diffusion model, conditioned on segmentation map and known pixels.

**POVNet: Image-Based Virtual Try-On Through Accurate Warping and Residual (Architecture)**

**Semantic Layout Generator (SLG):**

-Architecture: U-Net

-Input: Neutral garment image, initial incomplete layout, and pose

-Output: Complete layout

-Training Data: Pairs of neutral garment images and images of models wearing those garments

-Loss Functions: Pixel-wise cross-entropy loss and LSGAN loss

**Multi-Warp Garment Generator (MGG):**

-Components:

-Warper: Aligns garment image with the semantic layout using distance transform features

-Inpainting Module: Generates the final image given all warps, semantic layout, and incomplete image

-Warper:

-Alignment Technique: Uses multiple coordinated warps instead of a single warp

-Feature Utilization: Utilizes distance transform features to control garment deformation

-Training: Trained with cascade loss, per-pixel L1 loss, perceptual loss, and Spectral Norm GAN with hinge loss

**Residual Enhancement:**

-Objective: Ensure generated images show garment detail at the highest available resolution

-Procedure:

-Computes the difference between the warped garment image and the generated image, masked using the binary mask for the garment type

-Applies median filtering to estimate missing high spatial frequencies

-Adds the filtered residual back to the generated image

-Implementation Options: Can be applied using a heuristic-based approach or through end-to-end training of a residual classifier network

**Implementation Details:**

-SLG and MGG are trained jointly, possibly in an end-to-end manner.

-Utilizes off-the-shelf human parsing models and pose estimation techniques for data preprocessing.

-Distance transform features are used in the warping process to control garment deformation.

-Differentiable training enables joint training of components, allowing for optimization across the entire network.

**Distance Transformation Feature:**

A distance transformation converts a binary digital image, consisting of feature and non-feature pixels, into an image where all non-feature pixels have a value corresponding to the distance to the nearest feature pixel.

Computing these distances is in principle a global operation.

**TryOnDiffusion:**

Base Diffusion Model: Utilizes a Parallel-UNet architecture with an input resolution of 128x128. It comprises encoder-decoder structures with skip connections and outputs noise predictions.

Super-Resolution (SR) Diffusion Models: Incorporates two SR diffusion models for different resolution transitions. The first uses a Parallel-UNet for 128x128 to 256x256, and the second employs an Efficient-UNet for 256x256 to 1024x1024.

Architectural Features: Employs adjustments in decoder parts for upsampling, noise conditioning augmentation, and possibly residual connections.

Training Details: Involves loss functions like weighted denoising score matching objectives, the use of the Adam optimizer, specific learning rate schedules, and data augmentation techniques such as random Gaussian noise augmentation.

**Virtual Try-On with Pose-Garment Keypoints Guided Inpainting:**

Pose-Oriented Garment Keypoints Prediction: Utilizes a two-stream Graph Neural Network (GNN) architecture for predicting garment keypoints based on pose information.

Segmentation Map Generation and Cloth Warping: Employs an autoencoder architecture for generating segmentation maps and a specialized warping architecture for aligning garment images based on pose-oriented keypoints.

Semantic-Conditioned Inpainting: Utilizes a denoising diffusion model conditioned on segmentation maps and known pixels for inpainting missing regions in the garment.

**POVNet: Image-Based Virtual Try-On Through Accurate Warping and Residual:**

Semantic Layout Generator (SLG): Adopts a U-Net architecture for generating complete layouts of garments based on neutral garment images, initial incomplete layouts, and poses.

Multi-Warp Garment Generator (MGG): Consists of a warper component aligning garment images with semantic layouts using multiple coordinated warps, and an inpainting module generating the final image.

Residual Enhancement: Implements a residual enhancement step to ensure detailed garment representation by computing the difference between the warped garment image and the generated image, adding the filtered residual back to the generated image.