Try Fit

Virtual try-on Egyptian brands

Presented by

Genyveyav Raafat Louka
Toqa Osama Ali
Hassnaa Hassan Saied
Maria George Kamel
Habiba Mohammed Yahia
Monica Adel Lotfy

Supervised by

Prof.Dr. Abeer Mahmoud
TA. Mohamed Essam

TABLE OF CONTENT

1- Introduction

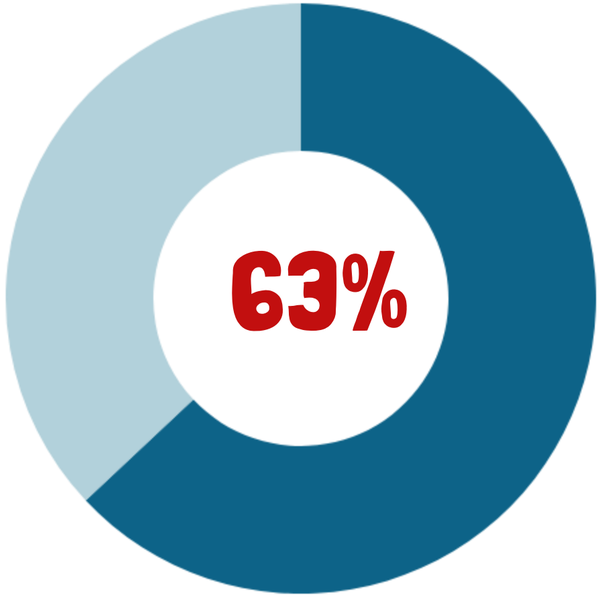
Introduction

TryFit is a multicategory project that brings together artificial intelligence, computer vision and fashion technology. It goes beyond traditional VTON by focusing on realistic garment fitting, cultural diversity and support for modest fashion.

2- Problem Definition and Motivation

Problem Definition

In Egypt's growing online fashion market, many shoppers struggle to assess clothing fit, leading to high return rates. For veiled women and privacy-conscious users, the lack of virtual try-on options adds to the difficulty. Going to stores is also time-consuming.



online clothes refund rate

Motivation

Motivated by these challenges, our team set out to build a solution that not only improves fit accuracy but also respects cultural preferences and enhances the online shopping experience in a personalized and intelligent way.

3- Objective

Developing an AI-powered, Virtual Try-On (VTON) system

Culturally Inclusive Try-On

Designed with hijab integration for diverse and inclusive virtual fitting experiences.

Support Local & Regional Fashion

Built to promote Egyptian designers and showcase homegrown brands with regional styles.

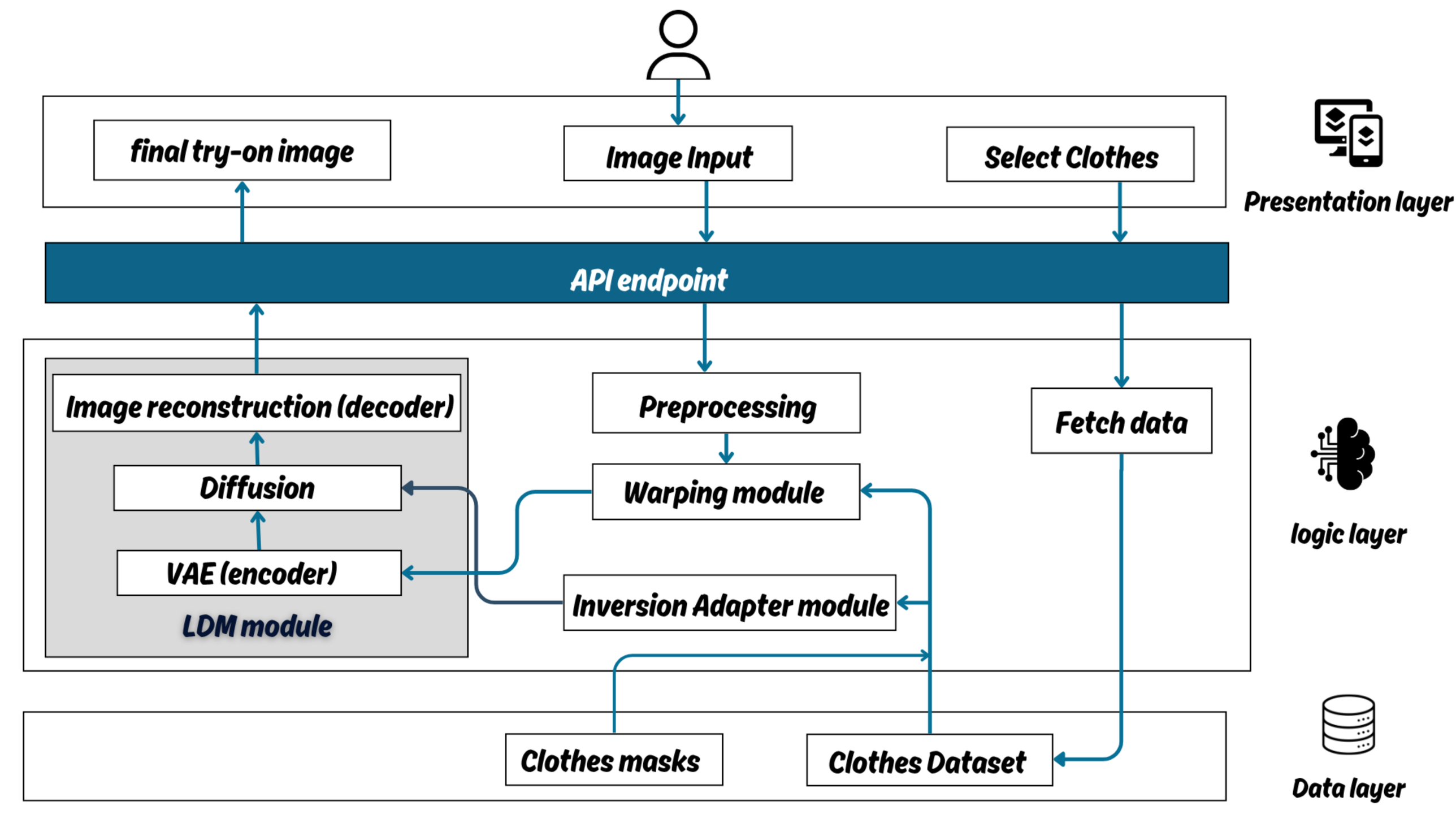
Reduce Returns

Focused on realistic fit visualization to improve accuracy and trust in online shopping.

Easy-to-Use Interface

Simple browsing, selecting, and trying on clothes with a seamless user-friendly design.

4- System Architecture


final try-on image
Image Input
Select Clothes
Presentation layer
API endpoint
Image reconstruction (decoder)
Diffusion
VAE (encoder)
LDM module
Preprocessing
Warping module
Inversion Adapter module
Fetch data
logic layer
Clothes masks
Clothes Dataset
Data layer

5- Dataset

Data




Mamzi


Saqhoute




Niswa


What's up

Description of Dataset

Name	Size	Description	Resource
TRYFIT Dataset	41 pair images (FULL) 93 pair images (LOWER) 55 pair images (UPPER)	**unveiled models**	**web scraping Local brands websites**
	141 pair images	**veiled models**	
	399 pair images (FULL) 523 pair images (LOWER) 663 pair images (UPPER)	**unveiled models**	**DressCode Dataset**
	1000 pair images (FULL) 865 pair images (UPPER)	**unveiled models**	**Anchor Dataset**

6- Phases Description

PHASE 1

DATA PREPARATION MODELS

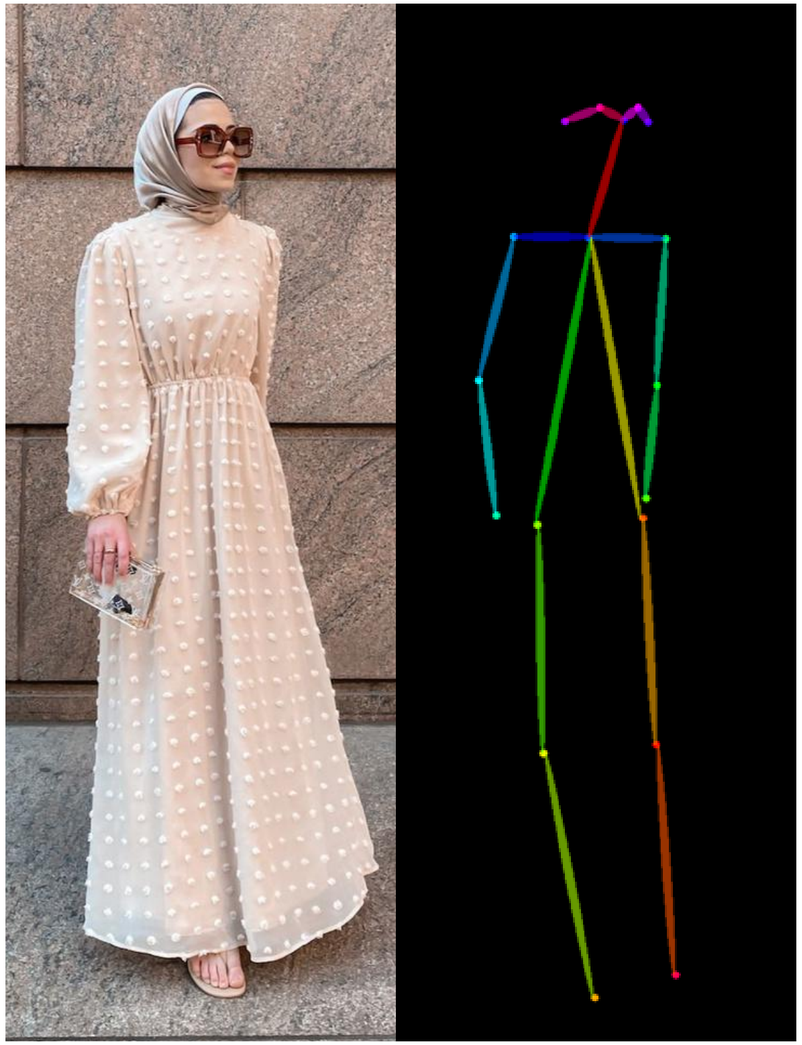
Segmentation

The model generates detailed human parsing maps that segment a person image into approximately 18 semantic regions (e.g., hair, face, upper clothes, pants, arms, and legs), which are then used to guide clothing warping and preserve identity-critical regions

Skeleton and Keypoints

We use pose estimation to extract (KEYPOINTS) that represent human joints and form a skeleton of the body.

This skeleton captures the person's pose and body structure, guiding accurate cloth alignment and warping.



Cloth Masks

Generated by extracting clothing regions from the image, producing binary masks where clothing areas are marked as 1 and non-clothing areas as 0 ensures accurate alignment and preservation of non-clothing regions

Face Masking

To reduce the model's focus on facial details and enhance attention on clothing, we applied a black face mask technique to the model image using its corresponding label map to localize the face region.

PHASE 2

TRY-ON PIPELINE

Warping module

-Warping module components serve as the essential first step

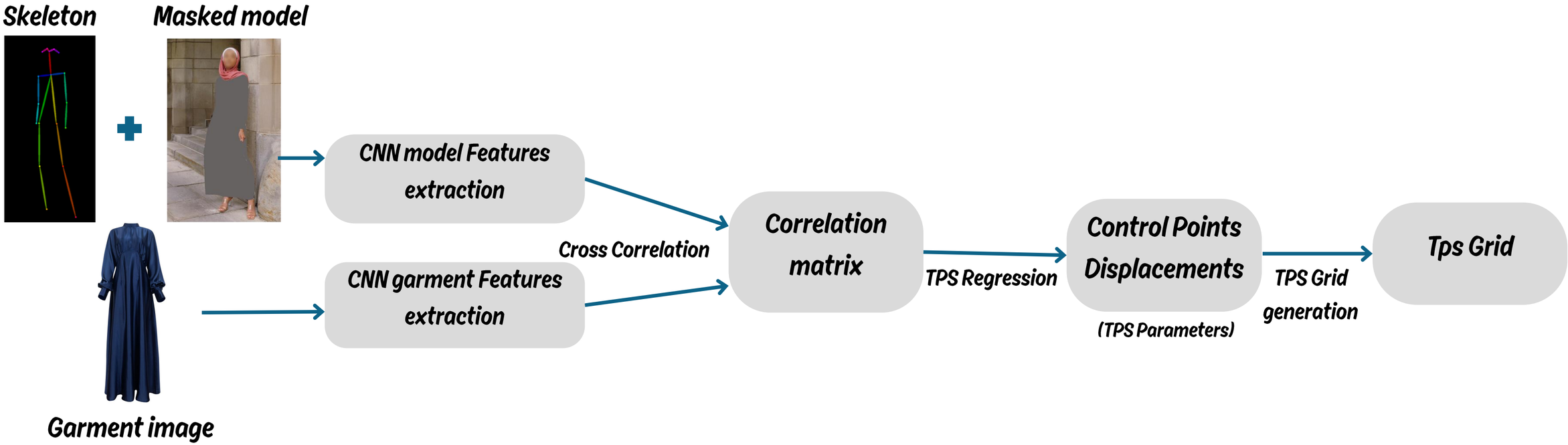
-Purpose : The Warping Module's job is to align the in-shop garment image with the body shape and pose of the target person. It ensures that the clothing looks like it naturally fits the person

-Warping Module components :

Warping module

1-Geometric Matching Module (GMM)

-Model : ConvNet_TPS – a convolutional network that predicts control-point displacements through Thin-Plate Spline (TPS) which is a smooth, flexible mathematical method for cloth warping.



Warping module

2- Warping Function

[
[[(-1.0,-1.0), (-0.33,-1.0), (0.33,-1.0), (1.0,-1.0)],
[(-1.0,-0.33), ..., , (1.0,-0.33)],
[(-1.0,0.33), ..., , (1.0,0.33)],
[(-1.0,1.0), (0.33,1.0), (0.66,1.0), (1.0,1.0)]]
]

Tps Grid

Garment

Coarsely warped garment

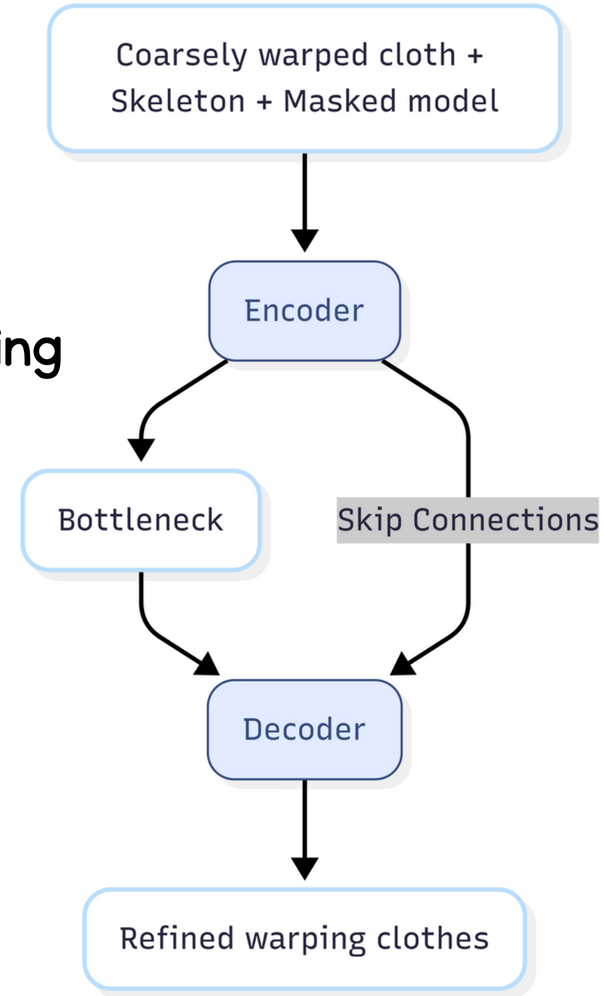
Warping module

3- UNet-Vanilla Refinement Network

-Purpose: Polishes coarse warped cloth output to enhance visual consistency and detail maintaining precise positioning while refining

-Key Components:

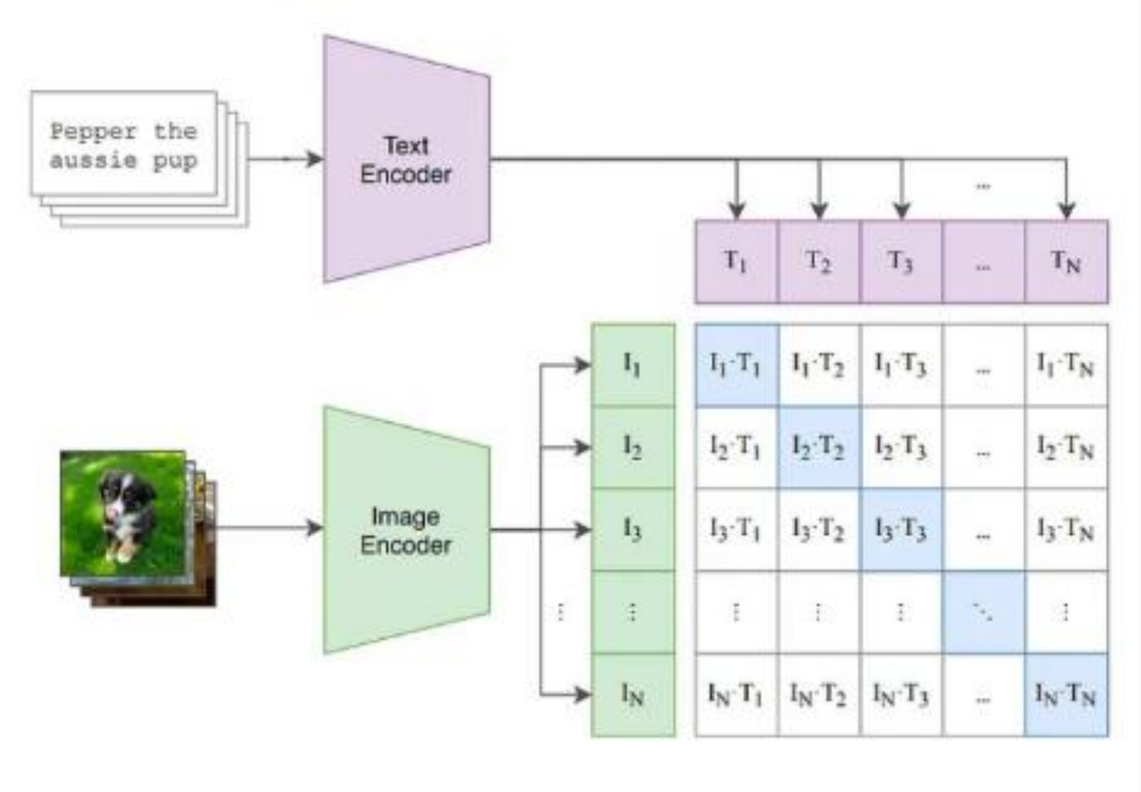
- Encoder (Downsampling)
- Skip Connections
- Decoder (Upsampling)



Inversion Adapter module

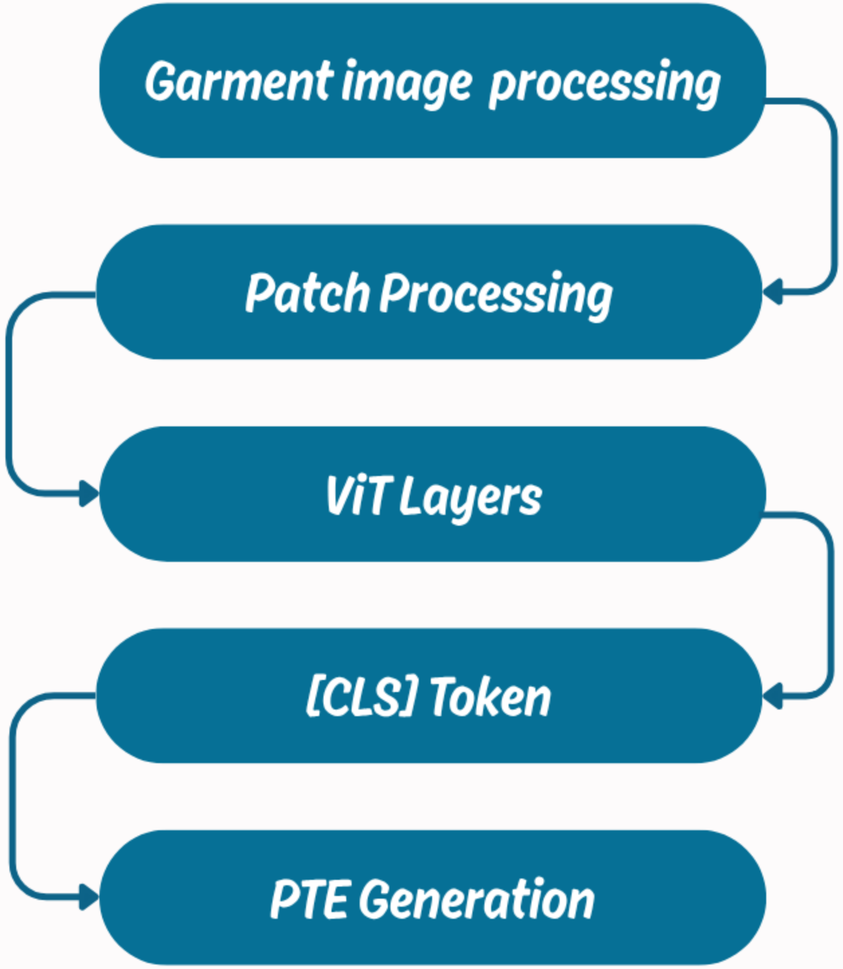
1- Vision Part Pipeline
(The Cloth Image)

2- Description part pipeline
(The Description)



1- Vision Part Pipeline (CLIP)

- The garment image is resized to 224x224 and normalized .
- Clip-ViT splits it into 256 16x16 patches.
- A [CLS] token is added to the patches.
- Transformer layers process the sequence.
- The [CLS] token captures global features.
- MLP converts it to PTEs.
- PTEs align visual features with text space.



2- Description part pipeline

1. Text prompt with placeholders
    a. (Ex: "A model wearing … $ $ $ $ blue dress")

2. Tokenization
    a. ["a", "model", "wearing", …,"$", "$", "$", "$", "a", "blue", "dress"]
    b. Map to token IDs: [320, 3293, 2674, 259, 259, …]
    c. Pad to 77 tokens: [320, …, 0,0]
    d. Getting tensor of shape [1, 77]

Blue dress

2- Description part pipeline

3. Embedding Lookup

- Each token ID ⟶ 768-dim vector
- Ex : "a" (ID 320) ⟶ [0.2, -0.1, ..., 0.5]
- Get input_embeddings tensor of shape [1, 77, 768]

4. PTE Injection

- Replace $ embeddings (positions 4-7) with PTEs from vision

Blue dress

2- Description part pipeline

5. CLIP Text Encoder

a. Self-attention links PTEs to descriptive words (e.g., "blue").

b. Output: Conditioned_text_embeddings of shape [1, 77, 768]

The Output of the module is Conditioned_text_embeddings of shape [1, 77, 768] as 768-dim vector for each token

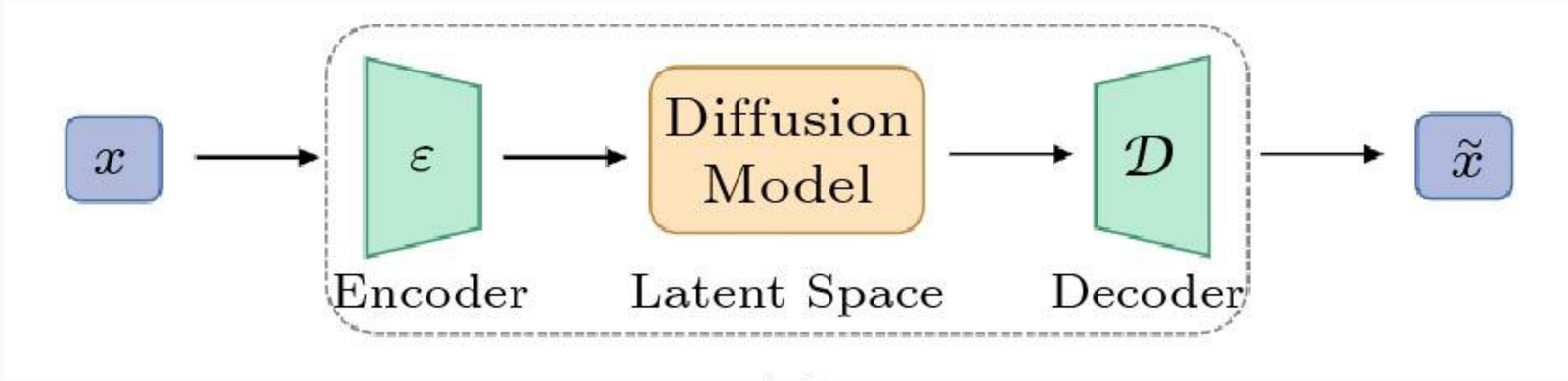
Conditioned _text_ embeddings

Token Position	Token Concept	Embedding vector (768)
0	[SOS]	[0.2, -0.1, 0.5, ..., 0.3]
1	"a"	[0.5, 0.0, -0.2, ..., 0.7]
2	"model"	[0.1, -0.3, 0.4, ..., -0.5]
3	"wearing"	[-0.2, 0.6, 0.1, ..., 0.4]
4-7	PTE1-PTE4	[0.8, -0.5, 0.9, ..., -0.3] (visual traits)
8	"a"	[0.5, 0.0, -0.2, ..., 0.7]
9	"blue"	[0.6, -0.4, 0.3, ..., 0.2]
10	"dress"	[0.3, 0.1, -0.7, ..., 0.0]
11-76	[PDA]	[0.0, 0.0, 0.0, ..., 0.0]

Latent diffusion module

Purpose

Generating photorealistic try-on results by blending warped garments with the target person's body.



Inputs

Warped clothe image

Target body

keypoints

Label maps

text embeddings

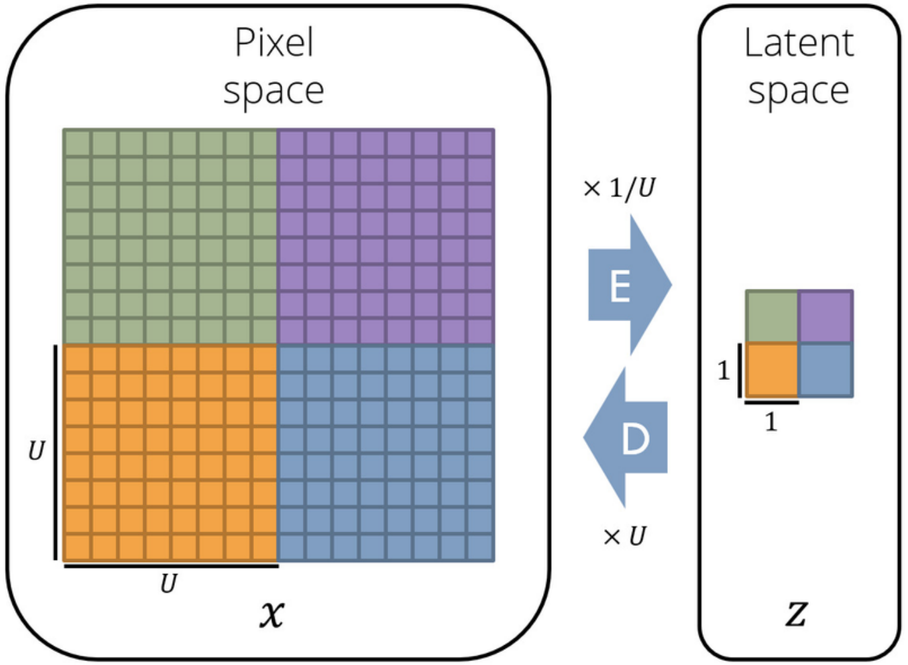
Output

Final Try_on image

Latent diffusion module

operates in a compressed latent space via VAE

Latent space is a compressed, lower-dimensional representation of data in LDMs



<u>Faster & lighter:</u>

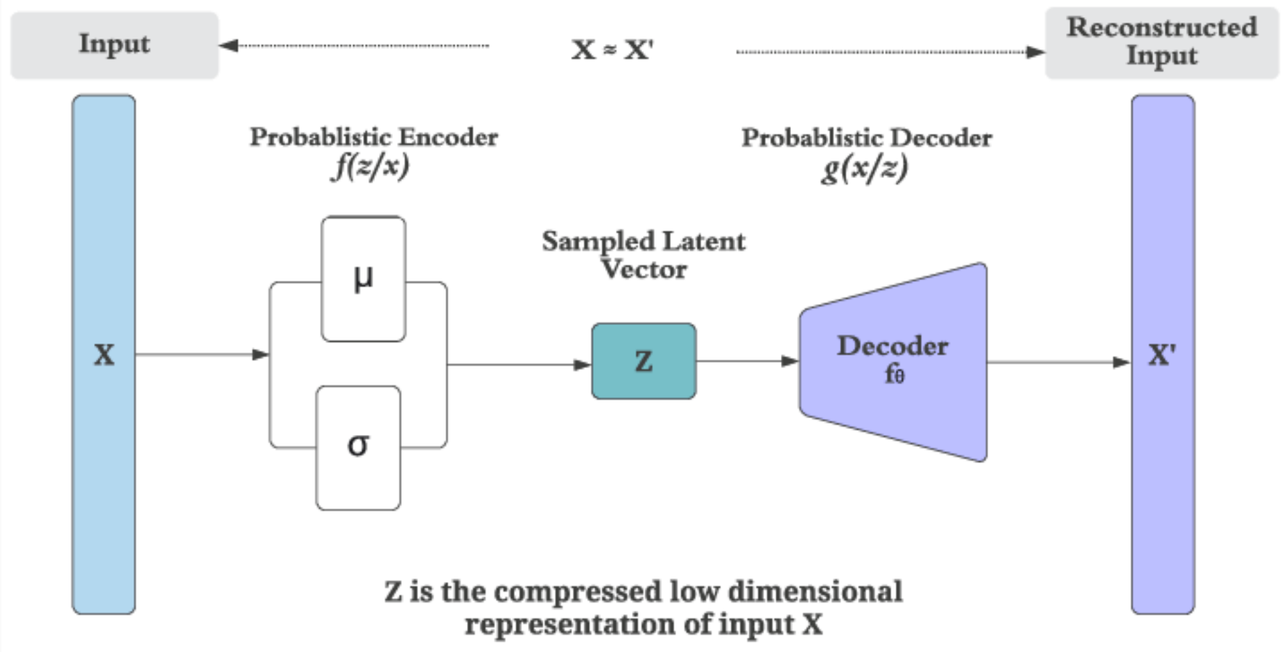
Uses 10-20x less memory by working in compressed space.

<u>Cleaner results:</u>

Focuses on key details by removing high-frequency noise

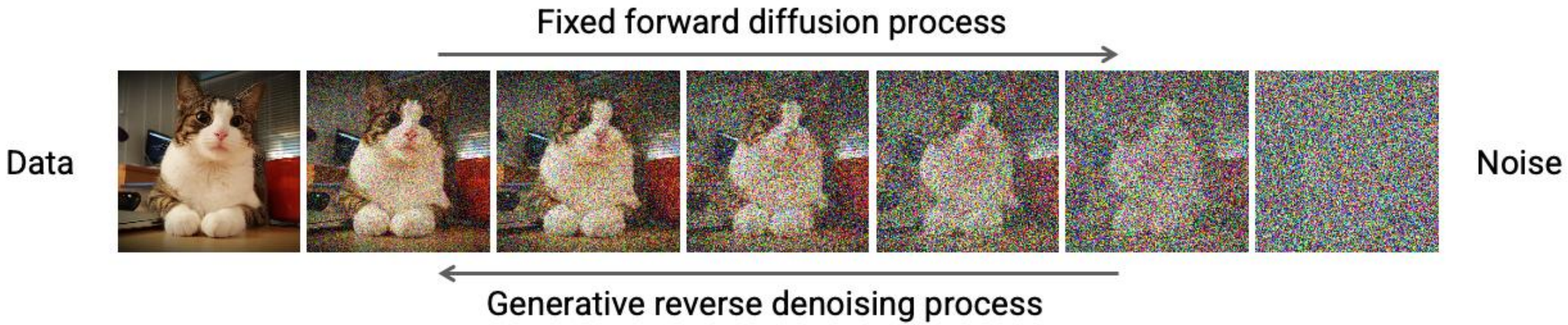
Variational autoencoder

VAE encodes high-dimensional data into a compact latent representation, reducing computational costs while preserving essential features.



Diffusion

Diffusion models are generative models that produce realistic outputs, especially in image generation, by using a step-by-step noise removal process.



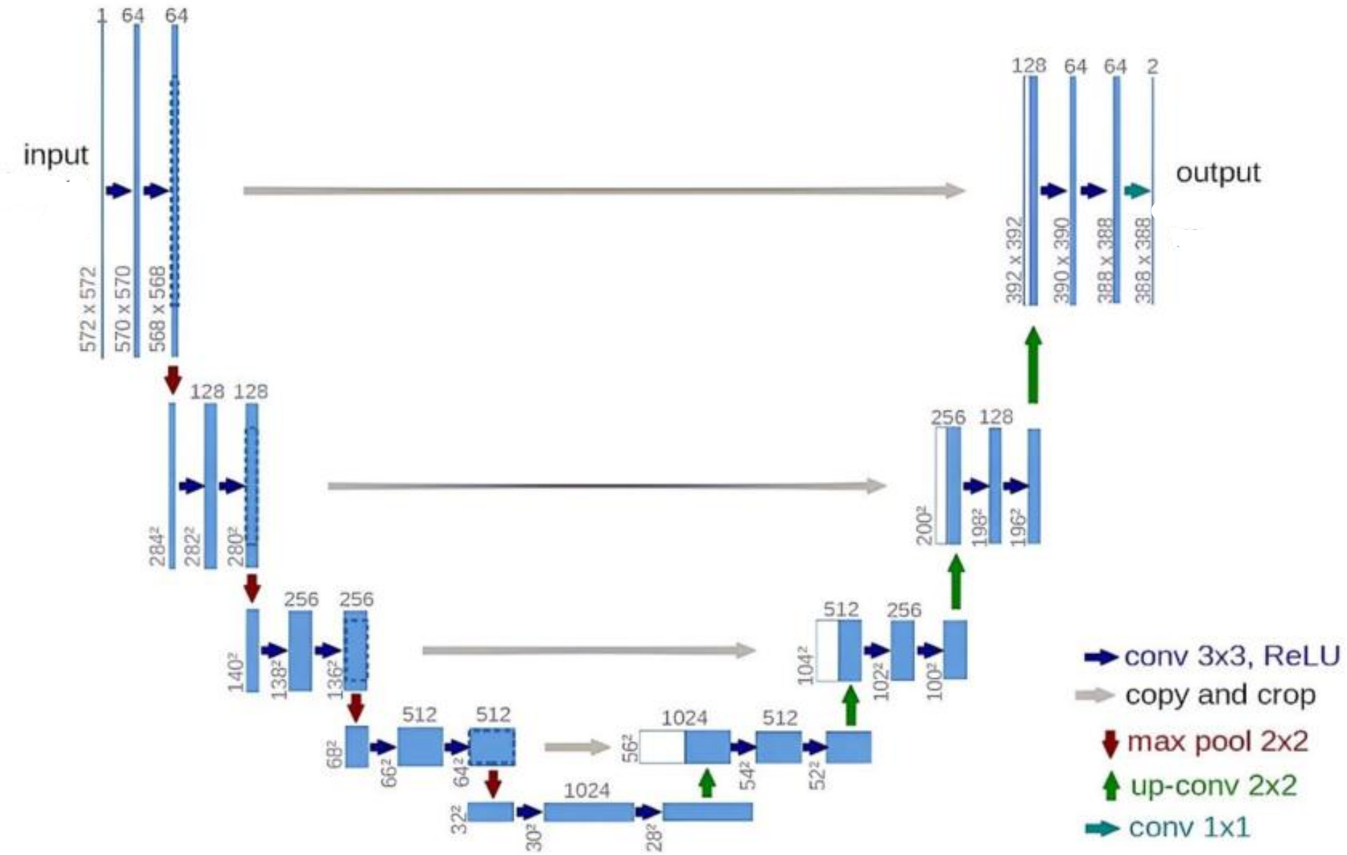
Forward Process: Gradually add noise to an image until it becomes pure noise

Reverse Process: Train a model to denoise step-by-step, reconstructing the original image.

UNet (Denoising phase)

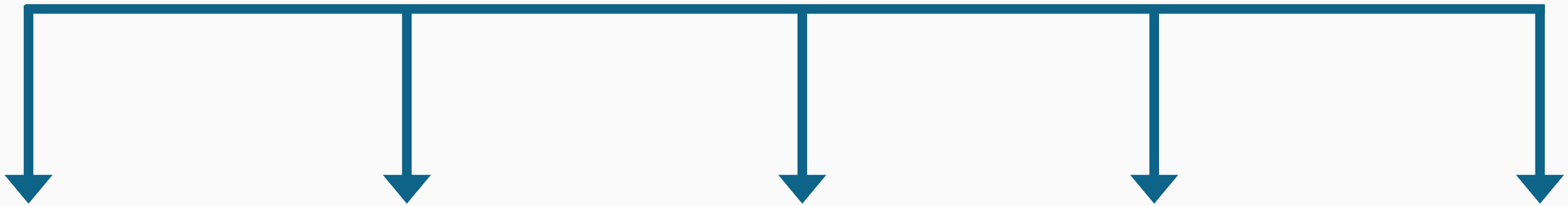
U-Net: The Noise-Predicting Workhorse

Predicts noise ε at each denoising step



UNet (Denoising phase)

The fusion happens inside the U-Net during denoising



Fusion process where multiple inputs are integrated to generate a realistic, pose-aligned, and semantically accurate try-on image.

EMASC

Enhanced mask aware skip connections works on the skip connections of the U-Net used for denoising.

1. Improves skip connections
2. Uses segmentation mask
3. Works at different scales to handle both small and big parts of the clothes
4. Fixes misalignment
5. Enhances detail and accuracy

LDM module flow

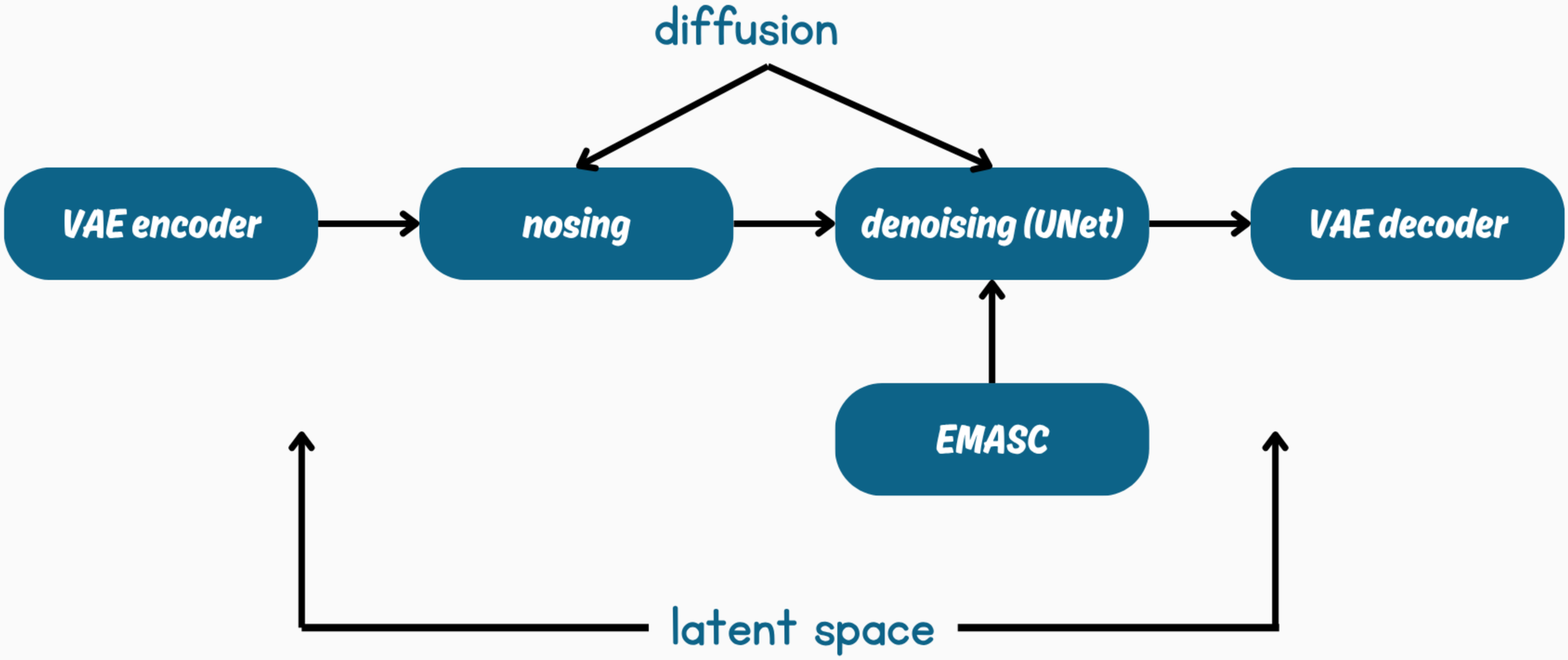


Image reconstruction


```
{
    "keypoints": [
        [
            152.5,
            58.5,
            0.9489192962646484,
            0.0
        ],
        [
            171.0,
            105.5,
            0.9177579283714294,
            1.0
        ],
```


[1,77,768]

CLIP text

embedding

7- Experiments and Results

Selected Try Fit Training Data

Dataset 1: 706 pairs of images / 121 veiled pairs

Dataset 2: 586 pairs of images / 10 veiled pairs

Dataset 3: 1274 pairs of images / 130 veiled pairs

Dataset 4: 1912 pairs of images

Evaluation Metrics Used

LPIPS (Learned Perceptual Image Patch Similarity) ↓	FID (Fréchet Inception Distance) ↓	SSIM (Structural Similarity Index Measure) ↑
Computes a distance based on deep features between a generated try-on image and the real (ground-truth) try-on image.	Measures how realistic and globally convincing and varied the generated image compared to real images	Pixel-level structural similarity (luminance, contrast, structure).
Together, LPIPS, SSIM and FID provide a complete evaluation. This balance helps assess both fidelity and visual quality in virtual try-on results.		

TRIALS

#	Dataset	EMASC steps	TPS epochs	refinement epochs	Inversion Adaptor	descriptions	Hyper parameter tuning	FID ↓	LPIPS ↓
1	dataset 1	2000	5	5	-	-	-	111.35	0.24
2	dataset 1	3000	5	5	-	-	-	78.49	0.147
3	dataset 2	3000	30	30	-	-	-	103.22	0.147
4	dataset 3	5000	50	50	✓	-	-	53.577	0.142
5	dataset 3	7000	50	50	✓	-	✓	53.283	0.140
6	dataset 4	7000	75	75	✓	✓	✓	57.219	0.168
7	dataset 4	7000	70	70	✓	✓	✓	59.249	0.169
8	dataset 4	7000	70	70	✓	-	✓	57.62	0.168

Final Result

	SSIM ↑	FID ↓
Try Fit model	**0.86825**	**53.2839**
ladi model	0.84894	58.3535

Dresses Visual Results

Garment	Body	Trial_1	Trial_2	Trial_4	Trial_5
					
					

Dresses Visual Results

Garment	Body	Trial_1	Trial_2	Trial_4	Trial_5
					
					

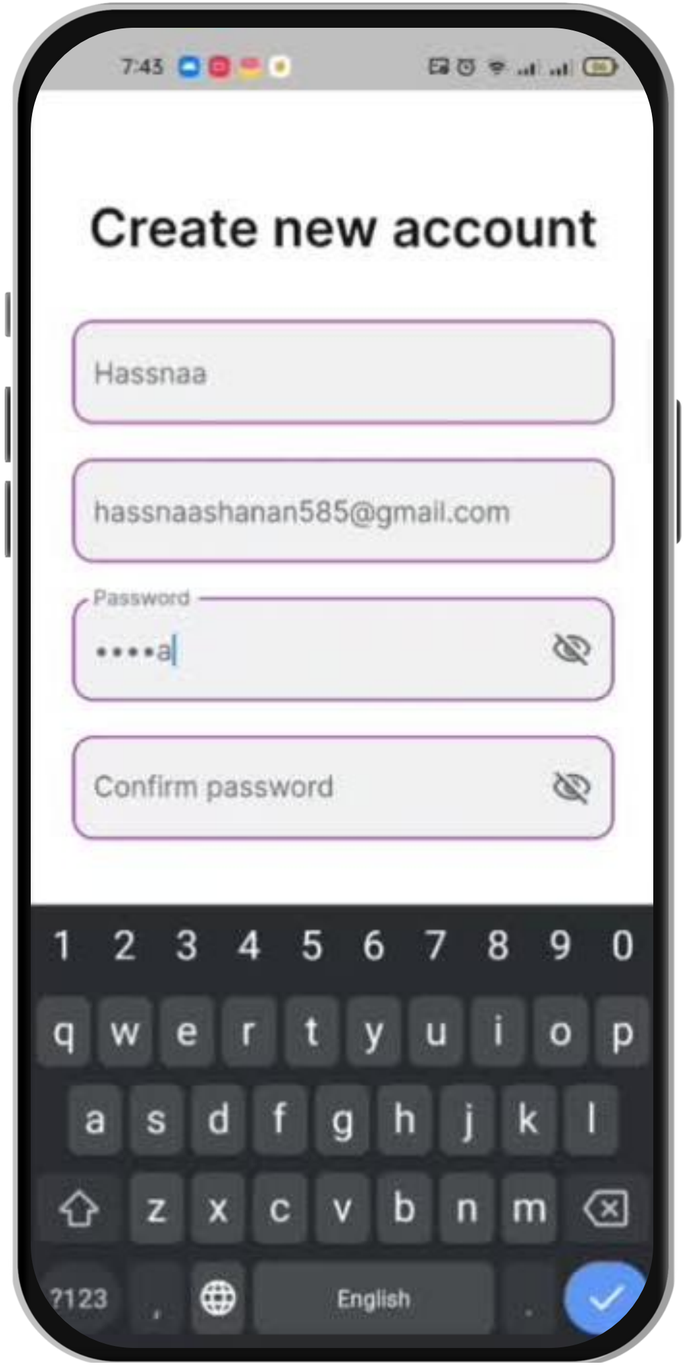
Upper body Visual Results

Garment	Body	Trial_1	Trial_2	Trial_4	Trial_5
					
					

Lower body Visual Results

Garment	Body	Trial_1	Trial_2	Trial_4	Trial_5
					
					

8- Demo


7:43
Create new account
Hassnaa
hassnaashanan585@gmail.com
Password
••••a
Confirm password
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
English

9- Conclusion and Future Work

Future Work

1. Multi-Layer Try-On
2. Recommendation System: Suggest items based on user preferences, wishlists, and previous try-ons to boost engagement.
3. Size Recommendation
4. Color Customization
5. Expanding Hijab & Modest Fashion Dataset

Conclusion

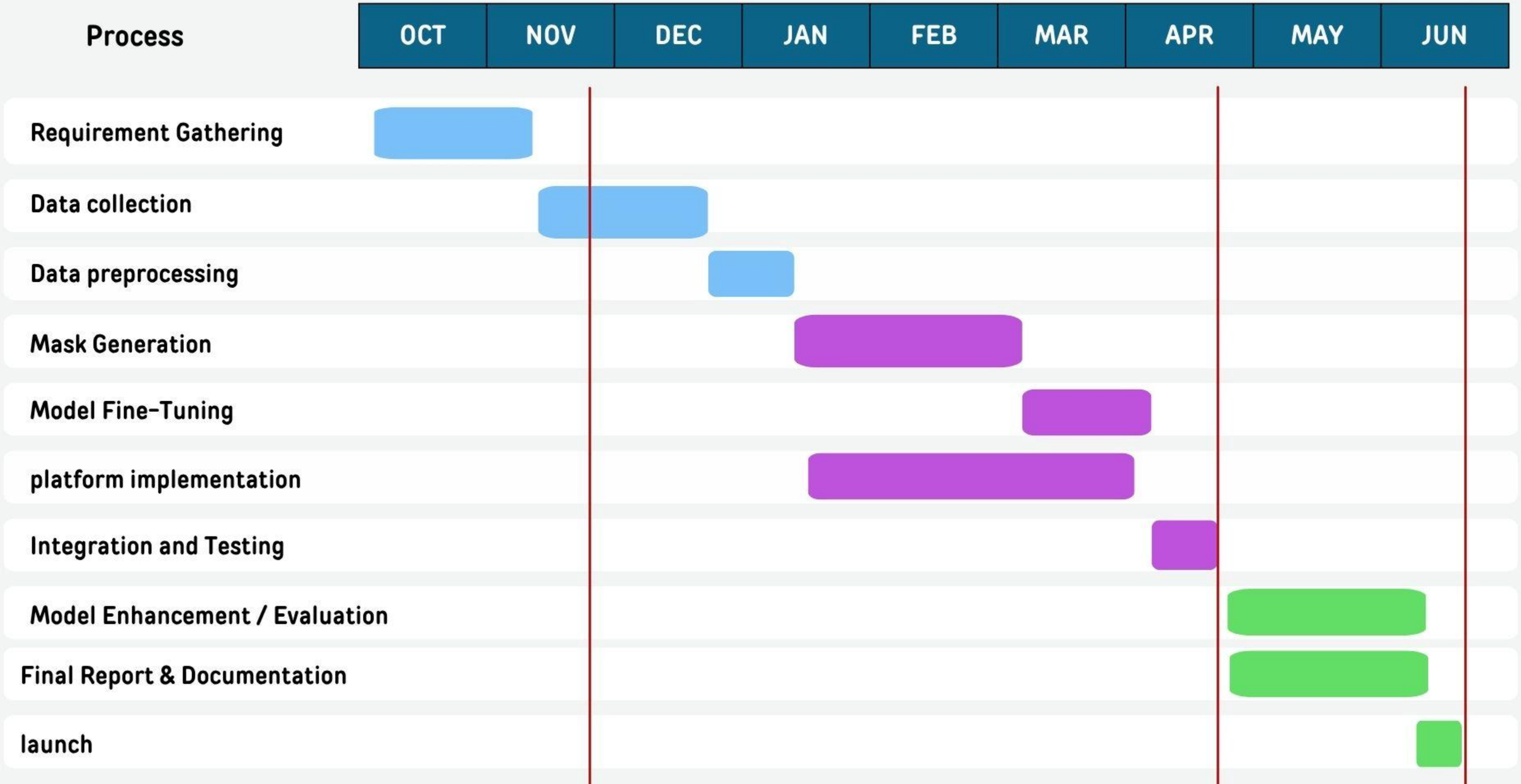
✅ We developed TryFit, a culturally inclusive Virtual Try-On system tailored for modest fashion and veiled users.

🧠 Integrated advanced deep learning models CLIP, TPS and EMASC to ensure realism.

📱 Built with a modular architecture using FlutterFlow, Firebase and Flask for seamless user experience and backend processing.

👗 Supports upper, lower, full-body garments – including hijabs – with high-quality try-on results.

Time Plan


Process
OCT
NOV
DEC
JAN
FEB
MAR
APR
MAY
JUN
Requirement Gathering
Data collection
Data preprocessing
Mask Generation
Model Fine-Tuning
platform implementation
Integration and Testing
Model Enhancement / Evaluation
Final Report & Documentation
launch

10- References

[1] D. Morelli, A. Baldrati, G. Cartella, M. Cornia, M. Bertini, and R. Cucchiara, "LaDI-VTON: Latent Diffusion Textual-Inversion Enhanced Virtual Try-On," Oct. 2023, doi: https://doi.org/10.1145/3581783.3612137.
[2] Z. Chong *et al.*, "CatVTON: Concatenation Is All You Need for Virtual Try-On with Diffusion Models," *arXiv.org*, 2024. https://arxiv.org/abs/2407.15886.
[3] S. Choi, S. Park, M. Lee, and J. Choo, "VITON-HD: High-Resolution Virtual Try-On via Misalignment-Aware Normalization." Accessed: Jun. 11, 2025. https://openaccess.thecvf.com/content/CVPR2021/papers/Choi_VITON-HD_High-Resolution_Virtual_Try-On_via_Misalignment-Aware_Normalization_CVPR_2021_paper.pdf.
[4] Y. Choi, S. Kwak, K. Lee, H. Choi, and J. Shin, "Improving Diffusion Models for Authentic Virtual Try-on in the Wild," *Lecture Notes in Computer Science*, pp. 206–235, Oct. 2024, doi: https://doi.org/10.1007/978-3-031-73016-0_13.

THANK YOU