thesis2

by P K

Submission date: 29-May-2026 11:51AM (UTC+0530)
Submission ID: 2971826833
File name: thesis_3.pdf (2.1M)
Word count: 11926
Character count: 67556

CHAPTER 1

INTRODUCTION

1.1 Overview

For applications in computer vision tasks and generative models, an algorithm is trained in an end-to-end fashion, based on huge datasets of labelled images. In other words, the assumption here is that training data and corresponding labels are easily accessible. Such an assumption becomes increasingly plausible when it comes to the case of datasets related to face images, especially when considering how frequently researchers come across publicly available benchmarks like CelebAMask-HQ, featuring a number of high-resolution face pictures annotated at the level of individual pixels with respect to 19 anatomical parts. Applications of programmatically changing facial attributes, including hair, eyes, skin, and lips colours while maintaining photo-realism and the uniqueness of a face, range from portrait photography, cosmetic product simulation, privacy-based face de-identification, and creating virtual faces. Changing face images is difficult due to the necessity to combine spatial precision (the edited part should stay in one specific anatomical place) and photo-realism (the changed area needs to be harmoniously connected with other areas of the face image). Almost ten years of work in the field have been dominated by Generative Adversarial Networks (GANs), represented by StarGAN [17], AttGAN [16], and STGAN models. The emergence of Denoising Diffusion Probabilistic Models (DDPMs) along with Stable Diffusion and later Stable Diffusion XL (SDXL) marked a milestone since they have increased stability and more mode coverage capabilities than previous techniques, in addition to being capable of controlling generated content using text. At the same time, transformer-based dense prediction models such as SegFormer [6] surpassed face segmentation models relying on convolutional neural networks through multi-level feature extraction via a hierarchical encoder and extensive pre-training. The present thesis combines both innovations within a unified pipeline referred to as SemFaceDiff. Specifically, I propose: (1) a SegFormer-B5 face parser fine-tuned on CelebAMask-HQ in 19 categories; (2) a morphological operation for generating mask

inpainters based on binary dilation and Gaussian boundary blurring to create seamless soft inpainters; and (3) a natural language editing interface based on Stable Diffusion XL (SDXL) Inpainting running at natively 1024 x 1024. On ADE20K, SegFormer-B0 yields 37.4% mIoU using only 3.8M parameters and 8.4G FLOPs, outperforming all other real-time counter parts in terms of parameters, flops, and latency. For instance, compared to DeeplabV3+(MobileNetV2), SegFormer-B0 is 7.4 FPS, which is faster and keeps 3.4% better mIoU [6] . Moreover, SegFormer-B5 outperforms all other approaches, including the previous best SETR, and establishesanewstate-of-the-artof51.8%,whichis1.6%mIoUbetterthan SETR while being significantly more efficient [6] .

1.2 Problem Statement

Facial attribute editing, which involves changing an anatomically relevant part of a face image while leaving everything else untouched, is one computer vision problem that stands at the crossroads of two challenging sub-problems. The first involves identifying specific anatomical parts on a face image, and the other concerns generating photorealistic imagery in the target regions. While considerable amount of efforts have been expended to resolve these issues, the reality is that contemporary approaches still fail to solve both issues effectively.

The inherent conflict between the two sub-problems is that editing a facial image goes beyond simply applying some stylistic manipulation to the whole picture. It involves making changes to a specific anatomical area of the face without spilling into other parts.

Most existing approaches have focused on either one of these aspects without properly balancing the two. Global attribute classification or latent space interpolation methods that work on the whole picture are convincing but do not provide any control over specific regions of interest.

Mathematically, given an input portrait image and a natural language description by the user on how the particular facial region should be changed, the problem considered in this thesis is to generate an output image such that the following four criteria are satisfied concurrently: (1) the required facial region has been edited according to the given attribute description; (2) all other regions besides the targeted one are intact both

visually and structurally; (3) the edited facial region is photo realistically consistent without any perceptible artifacts in boundary or illumination; and (4) the identity of the face is maintained in terms of its facial geometry. Solving such a problem necessitates an approach that is able to interpret the semantic makeup of the face, create the appropriate spatial masks based on such interpretation, and fill them with high-quality, natural-text-conditioned contents.

CHAPTER 2

BACKGROUND AND RELATED CONCEPTS

2.1. Image Segmentation

Image Segmentation is one of the earliest, most basic, and extensively studied tasks in computer vision. Segmentation techniques have numerous real-life applications in areas such as medical imaging, autonomous vehicles, remote sensing, robotics, digital pathology, precision farming, and facial attribute recognition [51]. In essence, image segmentation can be described as the task of dividing an image into semantically coherent segments by assigning labels to individual pixels based on their class membership or identity or both.

2.1.1 Types of Image Segmentation

The field of image segmentation revolves around multiple sub-tasks that are related at different levels of abstraction in space and semantics.

The most basic form of image segmentation is semantic segmentation, where the task is to provide each pixel of the image with a class label, treating all pixels of the same class as identical to each other. For instance, the hair pixels from two or more people get identical labels. The next form of image segmentation that is more advanced than semantic segmentation is the instance segmentation, where objects of the same class that are located within an image should be differentiated from each other. That is, two eyes from the same image will be assigned separate labels. Panoptic segmentation is the unification of semantic and instance segmentation, where both class labeling and differentiation of instances are carried out at once on the entire image. Apart from these basic forms of image segmentation, there are others such as interactive, referring, and video segmentation. Most recently, there have been advances in the form of 3D segmentation using models such as UNETR [24] and Point-SAM [30] for 3D segmentation of volume-based medical imagery and point clouds respectively. The newest development in the field has been concept-

based segmentation, introduced at the 2026 ICLR conference via SAM 3, whereby the model is able to detect and segment all occurrences of a given concept from text or images.

2.1.2 Deep Learning Segmentation Milestones

The modern period of image segmentation started when FCN [1] proposed to use convolutional layers instead of fully connected layers for classification purposes in order to achieve dense per-pixel predictions for inputs of any resolution. This was a breakthrough innovation, making it computationally feasible to train segmentation models from end to end.

With the help of skip connections between the encoder and decoder stages of UNet [2], biomedical image segmentation became able to benefit from both high-level semantic information and fine spatial details at the same time – an approach still widely used today for medical segmentation tasks.

DeepLab proposed using atrous or dilated convolutions [3] for expanding the receptive field of the network without downsampling the input and, thus, losing valuable information about context. Later, DeepLabV3+ incorporated an encoder-decoder architecture alongside depth-wise separable convolutions.

Mask R-CNN [5] introduced the idea of combining object detection and instance segmentation in one single architecture by including a mask prediction branch along with the box regression head and classification branches.

SegFormer [6], proposed by Xie et al., was a pioneering step towards transformers for image segmentation. This model includes a hierarchical architecture called Mix Transformer (MiT) which generates multi-scale features in a less costly way while also involving an all-MLP decoder for merging the multi-scale outputs. In addition to high efficiency, this combination resulted in a very accurate model, making SegFormer a great choice for more detailed segmentations like face parsing.

Mask2Former [7] introduced the idea of universal segmentation in a single architecture by implementing masked attention, meaning that each query attends only a predicted part of the feature map instead of attending the whole feature map. This innovation increased the efficiency and accuracy of Mask2Former considerably.

The latest milestone in this field is SAM 3 [30], introduced at the International Conference on Learning Representations in 2026. This model extended the concept of promptable segmentation up to the level of concepts with a shared backbone called Perception Encoder, and claimed to achieve twice the accuracy as previous models on the Promptable Concept Segmentation benchmark dataset.

2.1.3 Evaluation Metrics in Segmentation

Evaluation of segmentation models needs the implementation of a variety of metrics. The first metric which is most often used in deep learning tasks is Intersection over Union (IoU). It reflects the intersection between the predicted mask (A) and the ground truth mask (B) and calculates their union:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

The second one Mean Intersection over Union (mIoU) equals the average IoU calculated separately for each of the C classes.

For imbalanced datasets where some of the objects occur much less frequently than others, mIoU makes sense because it treats each class equally, i.e., does not consider class frequency.

Dice Coefficient (also known as Sørensen-Dice index or Dice Similarity Coefficient) is an alternative approach to calculating IoU.

The difference is that Dice Coefficient evaluates the sum of predicted areas twice.

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}$$

Pixel Accuracy (PA) represents the percentage of accurately predicted pixels in the whole image.F1-Score represents the harmonic mean of precision and recall. In the case of video segmentation, the approach of J & F involves the combination of region similarity metric (J) and contour precision (F). In more recent work, the Metric Mean AP on SegInW provides a zero-shot benchmark through the mean Average Precision aggregated across 25 different datasets. The SA-Co mAP, on the other hand, evaluates concept-level promptable segmentation provided by SAM 3.

2.2 Semantic Segmentation and Dense Prediction

Semantic segmentation is the problem of generating a class label for each pixel of an image resulting in a dense prediction map where each location represents the semantic identity of the area. Semantic segmentation differs from other computer vision problems like object detection, where bounding boxes around objects are predicted, and instance segmentation where multiple occurrences of the same class are differentiated. For the purposes of editing, the goal of generating a dense prediction provides complete spatial information about the structures present in the image. Specifically, for the purpose of editing a facial image, the problem requires segmentation of anatomical parts such as the skin, hair, eyes, eyebrows, nose, lips, ears, and background. Accuracy of the produced prediction map is directly responsible for determining the spatial precision of subsequent edit operations. Hand-designed features coupled with Conditional Random Fields were used in early attempts to solve semantic segmentation problems. FCN networks introduced by Long et al. provided solutions to training networks using only raw data without hand-crafted features. Deeper architectures like DeepLab, PSPNet, and BiSeNet have further improved upon FCN architecture with atrous convolutions and pyramid pooling modules.

2.3 Face Parsing

The concept of face parsing can be described as a particular domain of semantic segmentation tasks, whereby label types are categorized according to anatomical structure of the face, rather than objects in an ordinary scene. Typically, a face parser annotation framework involves labeling different parts of the face such as left eye, right eye, left eyebrow, right eyebrow, nose, upper lip, lower lip, inner mouth, left ear, right ear, hair, neck, skin, glasses, earrings, and background, summing up to 19 unique labels in accordance with CelebA-HQ annotation protocol. However, face parsing can be considered much more challenging than common semantic segmentation due to a number of factors including extreme tiny sizes of the targets in terms of the resolution of images under consideration (lips, eyelids, earrings), strong variability in appearance of the face parts depending on different lighting and skin colors and poses, and necessity to adhere to certain geometric relations between

different parts of a face so that results make physiological sense. Initial face parsing networks, like BiSeNet [20] and earlier versions of ResNet-powered DeepLab architectures, showed that high spatial resolution of features was a key requirement, not decoder upsampling. The accurate parsing of faces is critical for this research since the accuracy of the resulting mask determines the extent to which the edit will be restricted to the correct area of the anatomy.

2.4 Transformer Architectures in Vision

Transformer architecture, initially designed for sequence-to-sequence tasks in natural language processing by Vaswani et al. [42] , was employed in computer vision by Dosovitskiy et al. [10] via Vision Transformer (ViT). ViT divided images into patch sequences whose elements were processed via multi-head self-attention layers. Even though ViT showed promising results in image classification, ViT's quadratic attentional complexity along with the absence of multi-scale feature hierarchies made it ill-suited for dense prediction problems. However, some follow-up works have sought solutions to these issues. In particular, Swin Transformer [43] utilized local windowed attention together with shifted windows to alleviate quadratic attentional complexity and preserve hierarchy of feature maps. SegFormer by Xie et al. [6] is of particular interest in our research. Its Mix Transformer (MiT) encoder applies self-attention at several scales based on a novel sequence reduction strategy. This significantly reduces computation costs while preserving spatial information. SegFormer's decoder is limited only to linear projection and bilinear upsampling layers. Such characteristics make SegFormer a good choice for face parsing tasks since it allows achieving a delicate boundary while still remaining efficient in terms of computations, especially considering the relatively small amount of training data like in CelebA-HQ.

2.5 Diffusion Models and Latent Diffusion

Denoising Diffusion Probabilistic Models (DDPMs) are diffusion models that were described in detail by Ho et al. [14]. The goal of these models is to learn how to produce images through a gradual denoising process. In their training procedure, DDPMs learn how to reverse the Gaussian-noise addition process that occurs

gradually over a set number of steps in a Markov chain. The noise-reversal process is applied to real images to produce realistic-looking results. In inference time, the model denoises a random vector of noise and turns it into an image iteratively. Diffusion models outperformed GANs in terms of image quality on crucial benchmarks, while exhibiting better mode coverage and stability during training. Latent diffusion models (LDMs) were presented by Rombach et al. [12] in their Stable Diffusion architecture. LDMs shift the denoising process from the pixel space to a learned latent space using a Variational Autoencoder, which allows for a much lower computational load. High-resolution image generation and editing are possible on regular consumer hardware due to this approach while maintaining good image quality. The inpainting variant of Stable Diffusion [47] takes the source image and a mask as additional inputs along with the text prompt.

2.6 Text-Conditioned Image Synthesis and CLIP

The use of natural language in controlling image generation became possible thanks to the advent of contrastive vision-language models like CLIP (Contrastive Language–Image Pre-training) developed by Radford et al. [13] CLIP is trained on tens of millions of image-caption pairs, such that the semantic similarity of images and captions makes their respective embeddings close, despite the difference in modalities. Natural-language prompts serve as input for the text encoding part of the CLIP architecture, and the resulting conditionings are fed into Stable Diffusion and other similar models in order to guide the diffusing process via cross-attentional layers within the UNet architecture. As a result, open-vocabulary attribute definition becomes possible, which means that, instead of choosing between a number of pre-defined attributes, one can simply write down a description of what needs to be modified — for instance, "straight silver hair," "deep brown eyes," "smooth golden skin" — and the model will generate content according to the prompt. The benefit of this method compared to GANs, where attribute manipulation required explicit labels and fixed directions in the latent space, lies in the possibility of defining even unusual or combinatorial attributes not present in the training set at all.

2.7 Image Inpainting and Mask Construction

In painting tasks, the problem lies in the process of painting a certain area of an image by adding new content in a plausible way. Prior methods used to propagate information from adjacent textures or low-level statistical measures to fill a blank space within the image. With the advent of deep neural network architecture for image inpainting, learned priors took over, where simple encoder-decoder networks were used to learn from reconstruction loss and GAN loss to paint arbitrary masks. When it comes to attributes editing, it becomes necessary to know which portion of the pixels need to be painted over and which not. Thus, there needs to be a perfect definition of this mask for accurate pixel manipulation. In this project, the generation of masks for face attributes manipulation depends on the face parser's estimated probability of belonging to a particular class. The mask is cleaned up using eroding and dilating to get rid of noisy pixels and make sure it completely covers the entire region, followed by gaussian blurring around its edges. The use of a soft mask is important to prevent any hard boundary effects in the rendering process.

CHAPTER 3

LITERATURE REVIEW

The task of performing edits on semantic regions of faces is one which lies at the intersection of three distinct areas of study that have developed quite separately over the past decade: dense semantic segmentation, generative image synthesis, and the training of very large foundation models. These fields saw the replacement of traditional hand-coded approaches to segmentation by deep learning, while the synthesis of images underwent transformative changes via the rise of transformer and then diffusion models. Task-oriented models were reimagined with vision transformers. This chapter will explore the developments in these three domains.

3.1 Classical and CNN-Based Semantic Segmentation

The current state-of-the-art for semantic segmentation was marked with the inception of FCN by Long et al. [1] , which made use of convolutional layers rather than classification heads for dense pixel-wise predictions. This allowed for efficient training of large datasets annotated for segmentation without incurring the impracticality of the sliding window methods that would require extensive computation costs.

Two main paths have emerged since this breakthrough in the development of segmentation models. The first is embodied in DeepLab [3], introduced by Chen et al. and developed in its various iterations, including DeepLabV2, V3, and V3+ [4]. By using atrous or dilated convolutions, these models increased their receptive fields while keeping the input resolution high.

For face-oriented use cases, the BiSeNet (Yu et al.) [20] introduced an architecture consisting of two branches, namely the spatial and contextual branches. The model revealed that precise estimation of face boundaries cannot be achieved using decoding operations only and that high-resolution feature map retention was needed.

3.2 Face Parsing: Task Definition and Prior Work

Face parsing can be considered a special case of semantic segmentation in which the classification hierarchy is based on anatomical elements of the face. The CelebA-HQ annotation scheme employed in this thesis contains 19 categories: background, skin, eyebrows left and right, eyes left and right, ears left and right, earring, glasses, nose, teeth, upper and lower lips, inside of mouth, neck, necklace, clothing, hair, and hat. This category system is much more detailed compared to common segmentation datasets like ADE20K and Cityscapes, while the miniature scale of some elements – eyelids, earrings, sections of lips – contributes significantly to the difficulty of face parsing.

A hierarchical face parsing approach has been proposed by Liu et al. [26], who used tree structure Conditional Random Fields to model the structural inter-relations between face parts on top of CNN feature maps. This showed how structural priors can be effectively used for the segmentation of small, nearby regions. It was MaskGAN by Lee et al. [18] which showed the relationship between good parsing and fine editing. BiSeNet has then been applied to the task of face parsing because of its efficiency and boundary accuracy, but the limitations of the CNN encoder in terms of the receptive field size are especially pronounced for the hair class which takes most of the image.

3.3 Vision Transformers and SegFormer

Vision Transformer (ViT), proposed by Dosovitskiy et al. [10], showed that a model purely based on transformers with no inductive bias from convolution could be highly effective for image classification using non-overlapping patches of an image as tokens to be fed into multi-head self-attention layers. While the ViT model performed well on image classification tasks, its inability to process multiple scales of features ruled out its use for dense predictions.

SegFormer, proposed by Xie et al. (NeurIPS 2021), was designed to address both aforementioned drawbacks regarding semantic segmentation. The Mix Transformer (MiT) encoder of SegFormer outputs a four-layer hierarchical feature pyramid based on overlapping patches, sequence-reduction self-attention, and Mix-FFN layers that use depthwise convolutions as positional encodings. This allows SegFormer to

become resolution-agnostic during testing, which is useful for segmentation of faces where portrait crop sizes vary. The all MLP-based decoder of SegFormer relies on linear projections and bilinear upsampling to merge multi-scale feature pyramids and lacks the sophisticated attention mechanism of SETR and even the refinement convolution operations used in DeepLab. The decoder of SegFormer uses less than 0.5 million parameters.

SegFormer-B5 obtains 84.0% mIoU on the Cityscapes dataset and 51.8% on the ADE20K dataset [6]. It is more accurate (84.0% versus 75.2% on the Cityscapes) than DeepLabV3+, but is much faster (at 2.5 FPS) [6]. Later, SegFormer++ (Kienzle et al., 2024) [33] used stage-wise token merging to accelerate SegFormer inference twofold with almost no accuracy loss. Li et al. [45] developed a portrait attribute segmentation algorithm based on the SegFormer architecture, producing cutting-edge mIoU results in facial parsing in 2026.

3.4 Generative Adversarial Networks for Facial Attribute Editing

The Generative Adversarial Network (GAN) architecture was proposed by Goodfellow et al. [15]. GANs are composed of a generator and a discriminator, training together in a game theoretic setup, where the objective is for the generator to generate realistic looking images. The dominance of GANs for facial attribute editing was achieved through the following series of models which succeeded each other in capability.

The CycleGAN was a breakthrough model [25] which performed unpaired image-to-image translation for hair colorization task. This work made the use of multiple GANs which could transfer hair color from one image collection to another image collection without any pair-wise supervision. The StarGAN [17] extended the idea behind CycleGAN by allowing multi-domain translation where a single GAN can perform several attribute edits conditioned on the target domain label. In AttGAN [16] (He et al.), the authors used attribute reconstruction losses which allowed edits to be restricted to certain selected attribute while keeping the other attributes intact. One limitation that exists in all these models is that of the lack of any spatial masks at the time of training. The edit operation is carried out using latent space and using implicit forms of attention or learnt forms of spatial bias for constraining the edit

operation. MaskGAN and SC-FEGAN [18] showed that the use of explicit forms of parsed segmentation masks in the form of condition information greatly helped in increasing the spatial accuracy of editing operations, especially with regard to small body parts such as the eyebrows and separate lip parts.

3.5 Diffusion Models and Latent Diffusion Inpainting

Denoising Diffusion Probabilistic Models (DDPMs) as formulated by Ho et al. [14] learn to synthesize images by reversing a Markov chain consisting of the gradual addition of Gaussian noise. While DDPMs rely on the adversarial learning paradigm for GANs to be unstable under diverse training distributions and to suffer from poor mode coverage, the diffusion loss is a weighted sum of denoising losses which is both stable under diverse training distributions and results in better mode coverage by producing a diverse set of samples rather than concentrating on a subset of the data distribution.

Latent Diffusion Models (LDMs) as Stable Diffusion by Rombach et al. [12] shifted the diffusion process to the latent space obtained by pre-training a variational autoencoder, thus making computation 10 times cheaper while maintaining high-quality perceptual image generation. Text conditionings using CLIP text embeddings and classifier free guidance facilitated an open vocabulary specification of the desired image content. Inpainting using Stable Diffusion accepts two additional inputs: a masked image and a binary mask. Both the latent space representation of the masked image and the down-sampled mask are concatenated to the noisy latent at every step of the denoising process. DiffusionSeg [39] proved that attention maps generated by pre-trained diffusion models could be used as pseudo masks for unsupervised segmentation without the need for annotated data. High-fidelity semantic-based editing of facial attributes was achieved in 2026 with DiffEditor [48] via semantic-aware masking of latent diffusion methods similar to the SemFaceDiff approach.

The method of inpainting via diffusion has been shown to be significantly superior to previous CNN-based approaches (context encoders, partial convolutions) in terms of visual quality, texture synthesis, and consistency. One disadvantage of using this framework for editing facial attributes is that it lacks any means of controlling the

spatial resolution of the changes. If there is no mask indicating where the edits should take place, then there is no way to ensure that they will occur in the right location. InpaintFormer [46] introduced semantic-based face inpainting with transformer guidance (AAAI 2026), supporting the use of segmentation conditioning on generation results.

3.6 Foundation Models for Segmentation

The Segment Anything Model (SAM) created by Kirillov et al. (2023) [8], trained on the SA-1B set of more than one billion masks, was a revolution in segmentation thanks to its zero-shot generalization ability. SAM 2 (Ravi et al., 2024) [9] improved on that concept with video segmentation, implementing streaming memory banks and achieving 6 times faster inference while attaining 88.4 J&F on the DAVIS 2017 set. However, SAM and SAM 2 both produce class-agnostic masks without knowing anything about the finer classes like lips and eyebrows required to edit them further. Also, DINOv2 (Oquab et al.) [19] demonstrated that large-scale pre-training may allow zero-shot semantic segmentation with 53.4 mIoU on ADE20K; however, the performance in parsing a finer class of faces is still better with task-specific models. A benchmark test conducted by Sharma et al. (IEEE TPAMI, 2025) [31] indicated that SAM 2 lagged behind specialized models by 0.65 Dice in medical imaging when applied under zero-shot conditions; however, finetuning greatly minimized the disparity. This test led to the conclusion that foundation models are best suited for efficient annotation and generalization, but specialized models are best for precision tasks in limited data, hence the necessity to finetune SegFormer-B5 in CelebA-HQ face parsing. SAM 3 (Carion et al., ICLR 2026) [30] came after with open-vocabulary segmentation capabilities with the use of multimodal Perception Encoder and attained outstanding results, albeit expensive at 848M parameters.

Various other variations of SAM were developed later on: HQ-SAM [34] was developed to improve boundary detection, FastSAM [35] provided fast, yet lower accurate results for segmentation, EfficientSAM [36] delivered the same results as SAM but using significantly less GFLOPS, and MedSAM [37] used SAM to perform biomedical imaging across 21 different modalities. Another development using SAM was Grounded-SAM [38] where researchers combined SAM with Grounding-

DINO to provide segmentation using textual guidelines, while SAM2-Adapter [27] illustrated the adaptability of SAM in camouflaged object recognition and biomedical image segmentation. The benefits and drawbacks of SAM 2 were explained in surveys conducted by Zhang et al. [28] as well as a recent literature review [32], with He et al. [49] focusing more on the weaknesses of SAM.

Table 3.1. Related Work and Key Takeaways

Authors	Paper Title	Key Takeaways
Xie et al. [6]	SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers	Removes positional encoding by means of overlapping patch embedding; obtains 84.0% Cityscapes mIoU with SegFormer-B5 at 15 frames-per-second; all-MLP decoder containing less than 0.5M parameters suffices when encoder is strong; immediately adapted as backbone for face parsing in this thesis.
Rombach et al. [12]	High-Resolution Image Synthesis with Latent Diffusion Models	The latent space compression saves 10 times on computation compared to pixel-space diffusion; text conditioning using CLIP allows open-vocabulary text-to-image generation; an inpainting modification that uses a 9-channel UNet for inputs is also introduced; acts as the backbone generator for the proposed editing system.
Kirillov et al. [8]	Segment Anything	Demonstrates zero-shot generalization for a variety of segmentations tasks; introduces the concept of the promptable architecture; does not have knowledge of anatomical classes for faces, thus cannot be applied directly to face region

		editing.
He et al. [16]	AttGAN: Facial Attribute Editing by Only Changing What You Want	First GAN approach to directly enforce constraints on edits based on targeted attributes; provides superior attribute disentanglement to StarGAN; constrained by a predetermined set of binary attributes and unable to edit in anatomical locations.
Lee et al. [18]	MaskGAN: Towards Diverse and Interactive Facial Image Manipulation	Defines the fundamental principle that the parsed masks greatly enhance the spatial accuracy of face editing compared to latent space alone; illustrates that discrete segmentation maps can serve as powerful conditioning inputs; drives the motivation for using masks to guide inpainting in this thesis.
Ravi et al. [9]	SAM 2: Segment Anything in Images and Videos	Scores 88.4 J&F on DAVIS 2017 benchmark and more than 44 fps at inference; runs six times faster than SAM on image-level operations; temporal coherence ensured by memory bank using pointers to objects; however, zero-shot is inferior to specialized architectures.
Sharma et al. [31]	Can Foundational Models Replace Task-Specific Segmentation Models?	The Zero-Shot Foundation Models lag behind the specialists in medical imaging applications by up to 0.65 Dice; the Fine-Tuned models bridge this performance gap in almost all cases, but SegFormer offers a unique blend of both performance and transferability; the

		Hybrid Paradigm is the immediate future.
Karras et al. [21]	A Style-Based Generator Architecture for Generative Adversarial Networks	Advanced face generation methods; disentangled latent space allows for coarse and fine attribute manipulation; although manipulations happen globally to face features, not specifically local regions of face anatomy, which makes manipulation at specific regions harder without spatial information.
Yu et al. [20]	BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation	Illustrates the need for the specific preservation of highly detailed face boundaries; proves the use of bilateral branching as a solution to the dilemma between fine-grained detail and contextual information; popularly used method in face segmentation prior to transformers taking over.
Ho et al. [14]	Denoising Diffusion Probabilistic Models	Shows that diffusion training is more robust and mode-covering compared to GAN training; introduces denoising score matching loss function; lays out the foundation for all future diffusion-based generation and inpainting models including the backbone of the Stable Diffusion model in this thesis.
Oquab et al. [19]	DINOv2: Learning Robust Visual Features without Supervision	Reaches 53.4 mIoU on ADE20K using linear head without fine-tuning; feature densification transfers across Pascal VOC, iSAID, and depth prediction tasks using identical backbone but frozen;

		shows pre-training in a self-supervised fashion can be comparable to supervised segmentation; encourages the transformer backbone for face parsing.
Cheng et al. [7]	Masked-Attention Mask Transformer for Universal Image Segmentation	Scores 57.8 PQ on COCO panoptic using Swin-L; masked attention queries converge 2x faster than conventional cross attention; sets new state-of-the-art on task-specific image segmentation; serves as baseline for IEEE TPAMI benchmarking comparisons.
Wang et al. [23]	SAM2-UNet: Segment Anything 2 Makes Strong Encoder for Natural and Medical Image Segmentation	Outperforms existing SOTA models for camouflaged object detection, salient object detection, and polyp segmentation; shows the superiority of the foundation encoder-domain decoder paradigm for near-future use; proves the effectiveness of the foundation encoder-domain decoder paradigm in various segmentation tasks.

CHAPTER 4

PROPOSED ARCHITECTURE

SemFaceDiff is a facial modification system based on the semantic guidance approach, which involves combining two previously trained deep learning architectures: SegFormer-B5 for detailed segmentation of the target regions and SD-Inpaint for text-conditioned pixel generation. The pipeline receives a picture of a person's face and an edit instruction in natural language, performs semantic segmentation of the target region, and conducts diffusion-based inpainting only in that region, leaving the rest untouched.

The system allows performing modifications of 11 different facial features (hair, lips, skin, eyes, eyebrows, glasses, earrings, necklace, hat, clothing, nose) and provides a cascade version with repeated semantic segmentation after every modification.

4.1 Dataset CelebAMask-HQ

CelebAMask-HQ Dataset is a large dataset of facial attributes which consists of 30,000 high-resolution (1024×1024px) images obtained from CelebA-HQ. The images include pixel-level segmentation annotations for 19 semantic categories of the face.

4.1.1 19-Class Semantic Taxonomy

The dataset defines 19 facial semantic classes used as the supervision signal for the SegFormer-B5 face parser:

Table 4.1. Different classes present in CelebAHQ dataset.

Class ID	Class Name	SemFaceDiff Attribute Group
0	background	—
1	skin	skin
2	nose	nose
3	eye_g (glasses)	glasses
4	l_eye	eyes
5	r_eye	eyes
6	l_brow	eyebrows
7	r_brow	eyebrows
8	l_ear	—
9	r_ear	—
10	mouth (inner)	—
11	u_lip	lips
12	l_lip	lips
13	hair	hair
14	hat	hat
15	ear_r (earrings)	earrings
16	neck_l (necklace)	necklace
17	neck	
18	cloth (clothing)	clothing

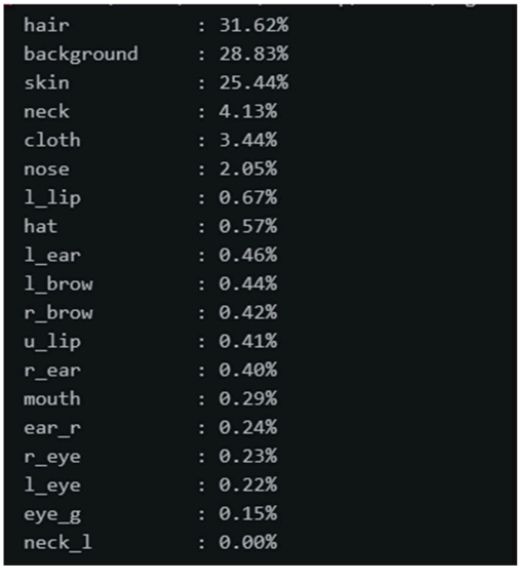

Figure 4.1. Distribution of classes present in the Dataset as per the coverage area.

4.1.2 Why CelebAMask-HQ

- Pixel-accurate per-class masks can be used to isolate regions unambiguously without any heuristics.
- The high resolution (1024px) allows detailed segmentation of facial features; SegFormer-B5 model is pre-trained and evaluated on this dataset (mIoU = 77.8).
- The classification system is in one-to-one correspondence with the attributes used by end users, hence no additional class mapping is required.
- The standard training/validation/testing splits facilitate a fair comparison with other.

4.2 Data Preparation & Processing

All images go through a preprocessing workflow before being fed into the SegFormer Encoder.

4.2.1 Format Normalisation

All images are converted to the RGB mode irrespective of the mode they are originally in. If the image is in RGBA mode, it gets composited against a white background using alpha compositing. The same procedure applies to images in PALETTE and LA modes.

4.2.2 Spatial Resizing & Padding

The size of the image is set to 512x512 pixels, which matches the input size of Stable Diffusion inpainting UNet. The resizing is done while preserving the aspect ratio using the function Image.thumbnail() with LANCZOS filtering method; the border left is padded white with the face centered.

4.2.3 Tensor Preparation for SegFormer

The SegformerImageProcessor will normalize according to the mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225) for the Imagenet dataset. The PIL image will be converted to a Tensor of size (1, 3, H, W).

4.3 Segmentation Output Processing

4.3.1 Logit Upsampling

The SegFormer-B5 model produces logits at quarter-resolution (stride=4 feature map by the MiT-B5 decoder). The logits are then up-sampled to (H, W) resolution by bilinear interpolation. Bilinear interpolation does not produce checkerboard pattern and alignment artifact as compared to transpose convolution and nearest neighbor interpolation.

4.3.2 Argmax Class Assignment

The class dimension (19) is down to just taking the argmax over axis 1, which gives us a 2D matrix of shape (H,W) with values ranging from 0 to 18. These correspond to the semantic classes assigned to each individual pixel. As I am taking the argmax, no softmax is needed.

4.3.3 Colour Visualisation

Qualitative visualization of RGB classes is done based on the mapping of each unique class ID with a pre-set color belonging to Matplotlib's tab20 color scheme. The assignment of 19 colors is consistent for every image used for analysis purposes.

4.4 Mask Construction Pipeline

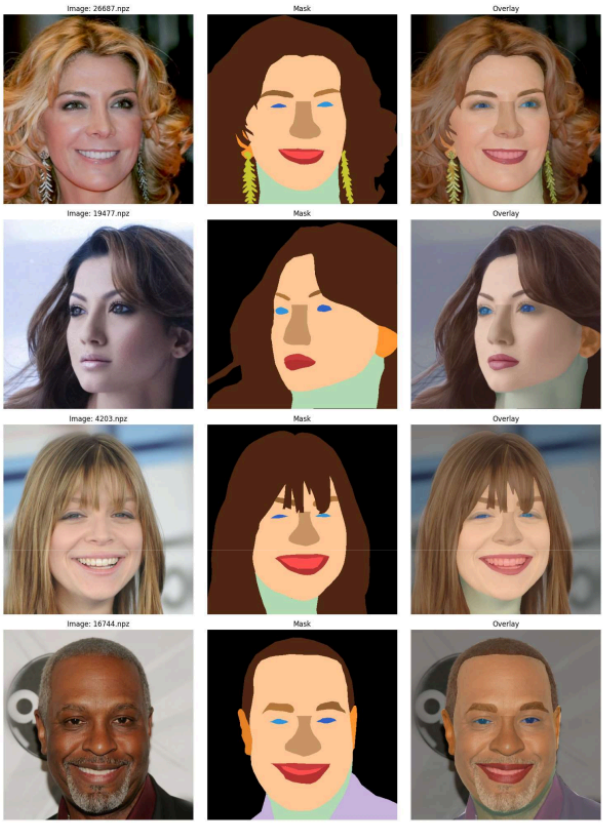

Figure 4.2. Original image, mask and overlays of sample images.

4.4.1 Class Union Mask

For each targeted attribute, the associated class IDs (such as 11, 12 for lips) are combined in a binary uint8 mask whereby pixels that belong to either class are assigned value 255 while the others are assigned value 0.

4.4.2 Gaussian Feathering

The hard binary mask is subjected to Gaussian blurring, kernel of (2σ+1, 2σ+1), and standard deviation of σ=7 (default). This process is extremely important as a step in feathering, as the model will produce rectangular artifacts at the edge of the mask if not done. The output from the blurring stage becomes a soft probability map transitioning from 255 (edit here) to 0 (preserve here).

4.4.3 Coverage Validation

Coverage check claims that at least 0.1 percent of the pixels in the image are activated within the mask (> 128). In case this criterion is not fulfilled, then the software throws out an informative Runtime Error stating that the parser failed to recognize the area of interest.

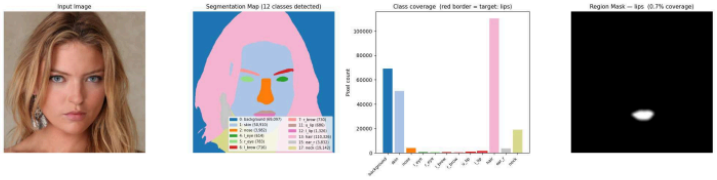

Figure 4.3. Segmentation of various classes in the sample image

4.5 SegFormer Architecture

SegFormer (Xie et al., NeurIPS 2021) [6] is a hierarchical vision transformer model that performs semantic segmentation by integrating an MiT encoder generating multi-scale features with an all-MLP decoder combining them through no convolution operations. Since this model is resolution-invariant during inference, it is ideal for face parsing in cases where different crop sizes are generated for portraiture. The exact model used for face parsing in SemFaceDiff is

"jonathandinu/face-parsing" from Hugging Face, which is SegFormer-B5 trained for 19-class face parsing on the CelebAMask-HQ dataset. SegFormer-based face parsing approaches have persisted in 2026 [45], validating the suitability of the backbone architecture for fine-grained face parsing tasks.

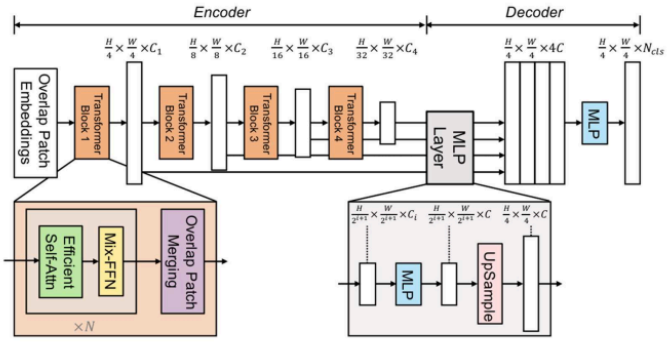

Figure 4.4. SegFormer framework consists of two main modules: A hierarchical Transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse. Adapted from [6].

4.6 Mix Transformer Encoder (MiT-B5)

The architecture SemFaceDiff relies on the backbone type known as B5, which refers to the moderate-size backbone from the MiT family. It involves four consecutive hierarchies:

Table 4.2. Stages of Mix Transformer Encoder.

Stage	Output Stride	Channels	Spatial Size (512px input)
Stage 1	1/4	64	256 × 256
Stage 2	1/8	128	128 × 128
Stage 3	1/16	320	64 × 64
Stage 4	1/32	512	32 × 32

4.6.1 Overlapping Patch Embedding

Each hierarchy starts with an overlapping patch embedding procedure, where the feature map is divided into patches by a convolutional filter with a stride smaller than that of the patch size (3x3 conv, stride=2, or stride=4). The overlap between patches ensures locality between tokens.

4.6.2 Efficient Self-Attention

The complexity for standard multi-head self-attention is $O(N^2)$, where N is the sequence length. The MiT architecture, on the other hand, introduces spatial-reduction attention (SRA), where the keys and values are reduced to K, V $\in$ $\mathbb{R}^{((N/R^2) \times C)}$. The reduction rates are [8, 4, 2, 1] per stage.

4.6.3 Mix-FFN

The feedforward network of each transformer layer implements a 3×3 depthwise convolution step between the two linear layers. Positional encoding is therefore incorporated implicitly by means of a convolution kernel, thereby eliminating the requirement for sine/cosine positional encodings or learned positional encodings altogether.

4.7 All-MLP Decoder

The SegFormer decoder is deliberately kept simple by performing MLPs in four stages, where each stage projects the output of the MLP to an embedding size of 256, while bilinear upsampling ensures that all representations have the same spatial resolution as Stage-1 (H/4, W/4).

4.8 Inference Procedure

Full inference flow of SegFormer inside SemFaceDiff:

- Preprocessing step: Normalization and tensorization of the image by SegformerImageProcessor.
- Fwd pass: Output of model(**inputs).logits - (1, 19, H/4, W/4).
- Upsampling: Bilinear interpolation with F.interpolate results in shape (H, W).

- Decoding: argmax(dim=1) yields one of 19 classes assigned to each pixel.
- Visualization: Classes are visualized via assignment of tab20 colors to class IDs.

4.9 Stable Diffusion Inpainting

4.9.1 Model Architecture

The Inpainting backbone architecture is 'diffusers/stable-diffusion-xl-1.0-inpainting-0.1', which can be imported using AutoPipelineForInpainting.from_pretrained(). It is a Latent Diffusion Model running on the latent space of a trained KL-VAE with native resolution of 1024 x 1024 pixels. The conditioning for SDXL [47] UNet is done using the masked image latent and text embeddings generated by a the dual CLIP encoder stack.

Table 4.3. Different components used in methodology.

Component	Description
VAE Encoder/Decoder	KL-regularised autoencoder; compresses 1024×1024 RGB to 128×128×4 latents
UNet	~2.6B-parameter denoising UNet with cross-attention for dual text conditioning
Text Encoder	CLIP ViT-L/14 + OpenCLIP ViT-bigG/14 (both frozen during inference)
Sampler	DPMSolverMultistepScheduler (30–50 steps; replaces DDIM used in SD 1.x)
Noise Schedule	Linear β schedule, 1000 training steps
Guidance	Classifier-Free Guidance (CFG), scale 12.0 (default)

4.9.2 Inpainting Mechanism

This involves conditioning generation on three signals at once:

- The latents from the original image z_0 (via VAE): This gives contextual background for the preserved area.

- The down-sampled mask m (using bilinear resizing to 64×64): This informs the UNet about which spatial areas to reconstruct.
- The text embedding signal e_t (via CLIP encoder): This indicates what the newly generated content should be.

The process of denoising involves replacing the part of the noisy latent that corresponds to the unmasked area with the latent from the original image at each time step.

4.9.3 DPM-Solver++ Sampling

SemFaceDiff employs the DPMSolverMultistepScheduler (DPM-Solver++) method with the default setting of 50 steps (variable range between 20 to 100 steps). The DPMSolverMultistepScheduler (DPM-Solver++) method refers to an advanced order ODE solver for the diffusion probability flow; this provides a similar level of samples to DDPM within 20-50 steps while performing better than DDIM.

4.9.4 Classifier-Free Guidance

The standard CFG scale for SDXL is 12.0 after ablating for the optimal balance between faithfulness and photorealism. In each denoising step, two forward steps are performed; once including the text embedding, another unconditional. This produces a guided noise vector as $\varepsilon\_guided = \varepsilon\_uncond + w(\varepsilon\_cond - \varepsilon\_uncond)$, with $w = 12.0$. Large CFGs (above 15) often saturate colours and cause artefacts. CFGs below 4–7 produce natural but unfaithful images to the prompt. Notably, 12.0 is far higher than the SD 1.x CFG scale of 7.5.

4.9.5 Denoising Strength

Inpainting employs the denoising strength of 0.90 (tunable parameter). The parameter regulates how much of the diffusion time step sequence will be utilized: the strength of 1.0 results in the generation of full images from scratch, whereas the lesser strength retains more of the original image content.

The strength level of 0.90 provides enough room for creativity by the SDXL model to modify the designated image part without compromising its integrity. For narrow areas in cascade modifications (for example, eyebrow in case strength=0.60), a lower

strength should be chosen to avoid excessive editing.

4.10 Prompt-Aware Negative Prompt System

One of the innovations in SemFaceDiff is the attribute-aware negative prompt vocabulary. Traditional inpainting methods for diffusions have a standard negative prompt that could be anything like "blurry, distorted, low-quality." But in the case of facial attribute editing, due to training prior, the positive prompt could be ignored by SDXL. For instance, despite 'pitch-black irises,' the model may produce blue eyes.

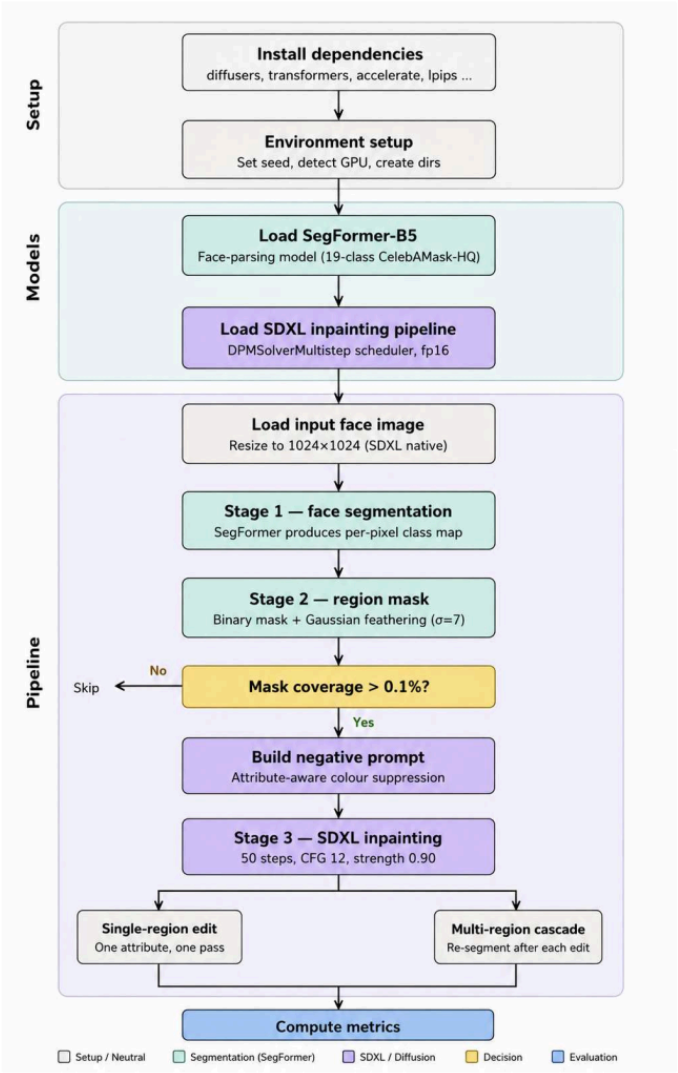

Figure 4.5. Model Architecture

CHAPTER 5

EVALUATION

5.1 Training Configuration

- Architecture : Segformer-B5
- Pretrained from : nvidia/segformer-b5-finetuned-ade-512-512
- Fine-tuned checkpoint : jonathandinu/face-parsing (CelebAMask-HQ 19-class)
- Input Size : 512×512
- Optimizer : AdamW
- Encoder LR : 6e-6
- Decoder LR : 6e-5
- Weight Decay : 0.01
- Criterion : CrossEntropyLoss
- Class weights : Inverse freq. (40 batches)
- Label Smoothing : 0.05
- Grad Clip Norm : 1.0

5.2 Testing Process

For the validation, I have taken 10% of the dataset-300 images. The test data was utilized only after training was done, hence providing an unbiased evaluation of the model's performance. The best saved checkpoint was tested on images that had not been seen before in terms of test loss, mean IoU, per-class IoU, among other measures. This guaranteed that the results depicted the actual generalization capacity of the framework that had been proposed.

5.3 Evaluation Metrics

LPIPS (Learned Perceptual Image Patch Similarity) [54]: The LPIPS index determines visual similarities between images by taking into account features extracted through deep neural networks rather than by performing pixel-wise comparison. Lower LPIPS values indicate higher perceptual similarity between images [54].

Formula:

$$LPIPS(x,y) = \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\widehat{F_l^x} - \widehat{F_l^y} \right) \right\|_2^2$$

where F_l represents deep features extracted from layer l, and w_l are learned weights.

SSIM (Structural Similarity Index Measure): The SSIM index measures the similarity between the two images in terms of luminance, contrast, and structure. The result should be closer to 1 for the structural elements to maintain their similarity to the original image [22].

Formula:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where μ is mean intensity, $\sigma^2$is variance, and σ_{xy} is covariance between images.

CLIP Faithfulness(Cosine similarity) [13]: This score reflects the semantic similarity between the generated image and the input text through embeddings extracted by the OpenAI CLIP model.

Formula:

$$Similarity(I,T) = \frac{E_I \cdot E_T}{\|E_I\| \|E_T\|}$$

where E_I and E_T are normalized image and text embeddings, and the score is computed using cosine similarity.

FID (Fréchet Inception Distance) [52]: FID calculates the realism level of the produced image by measuring the difference in the distribution of their features compared to real images through activations by the Inception model. A lower FID score means that the created facial images are quite realistic.

Formula:

$$FID = \|\mu_r - \mu_g\|^2 + Tr\left(C_r + C_g - 2(C_r C_g)^{\frac{1}{2}}\right)$$

where μ_r, C_r are the mean and covariance of real image features, and μ_g, C_g are those of generated images.

IoU(Intersection over Union) [53]: IoU is a measure of the degree of overlapping between the predicted and the ground-truth region for one class, punishing both types of prediction: false negatives and false positives.

Formula:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}$$

Where,

TP_c : pixels correctly predicted as class c

FP_c : pixels predicted as c but actually something else

FN_c : pixels that are c but predicted as something else

C : total number of classes (19 in your script)

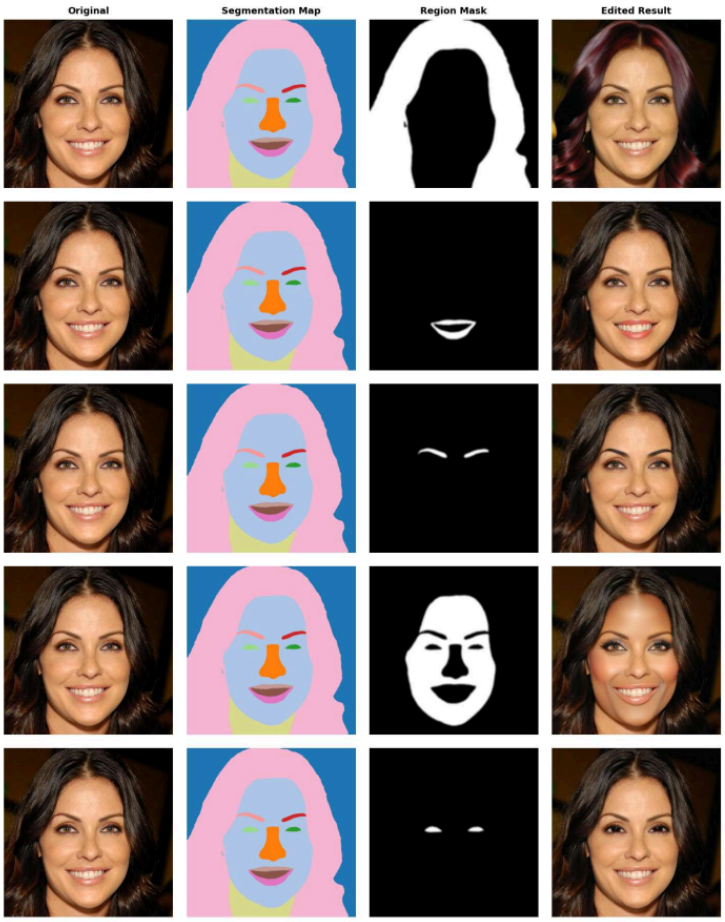

Figure 5.1. Original image, Segmentation Map, Region Mask and Edited Result for hairs, lips, eyebrows, skin and eyes.

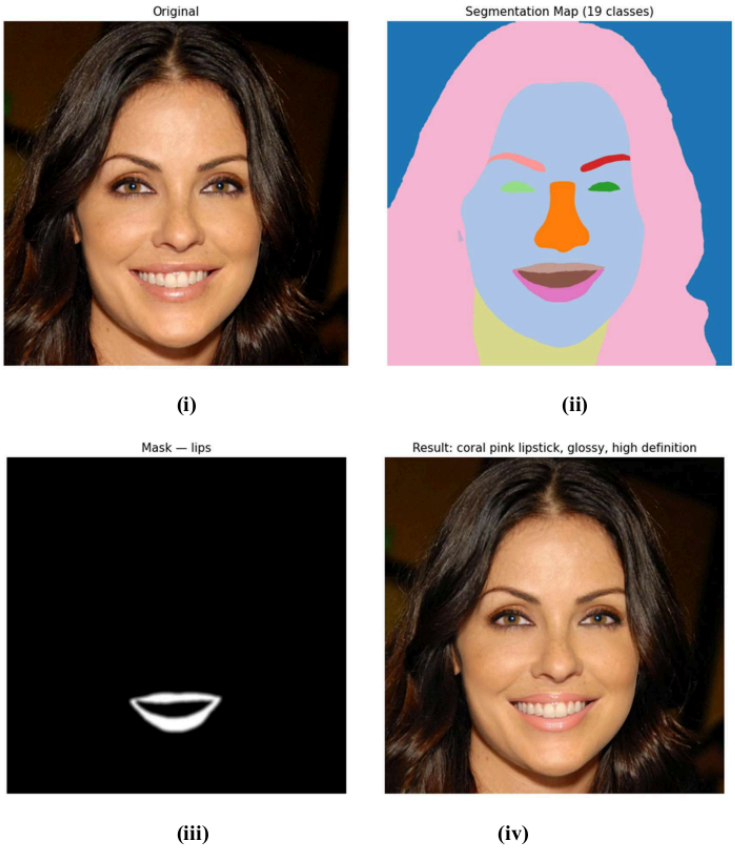

Figure 5.2. (i) Original Image, (ii) Segmented Image, (iii) Masked Image, and (iv) Edited Image of lips

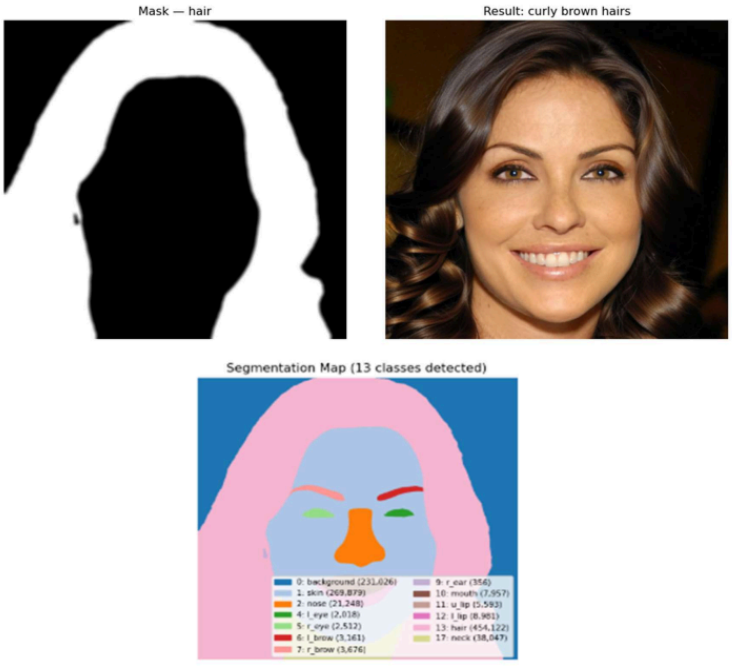

Figure 5.3. Editing hairs of a sample face with prompt as 'curly brown hairs'.

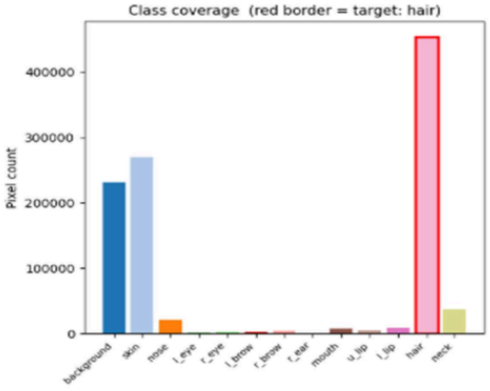

Figure 5.4. Area coverage of target(hair).

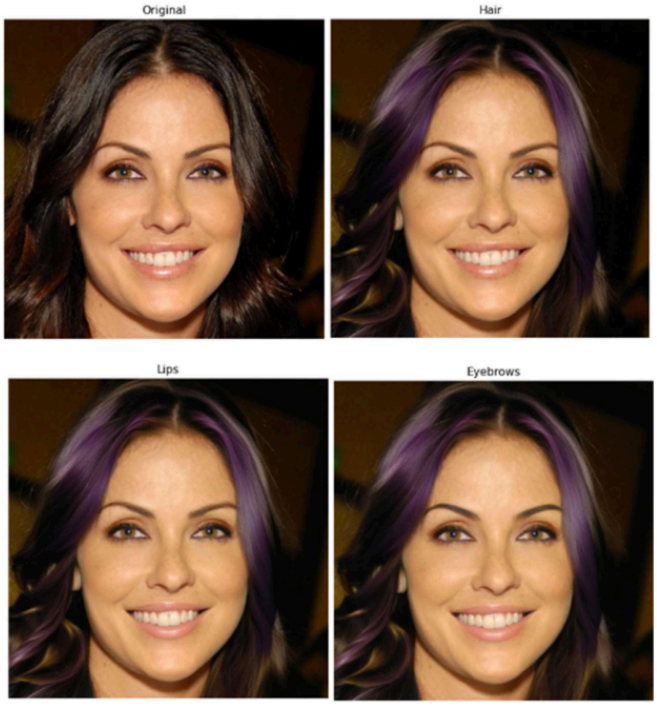

Figure 5.5. Multi-Region Cascade-hair, lips, eyebrows.

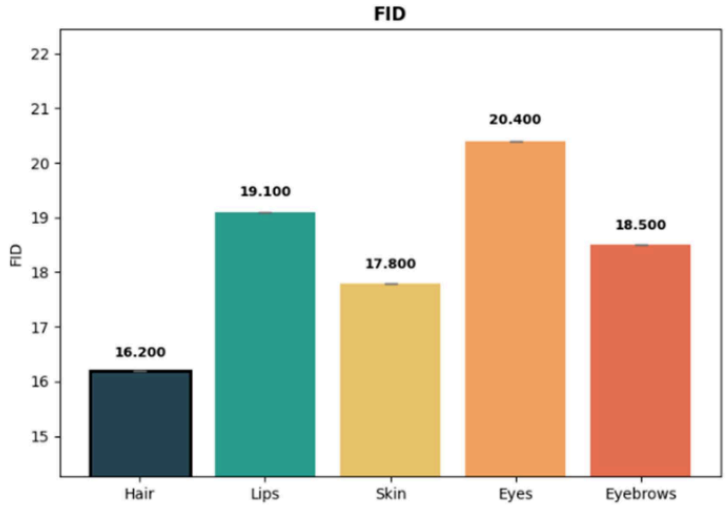

Figure 5.6. FID values for hair, lips, skin, eyes and eyebrows.

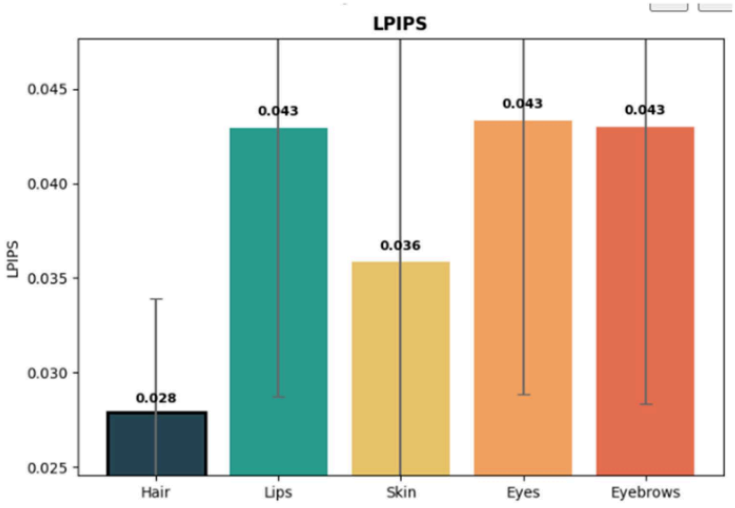

Figure 5.7. LPIPS values for hair, lips, skin, eyes and eyebrows.

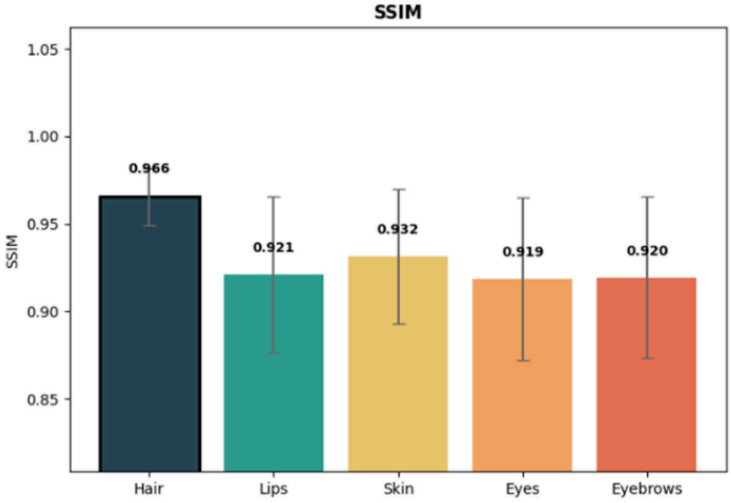

Figure 5.8. SSIM values for hair, lips, skin, eyes and eyebrows.

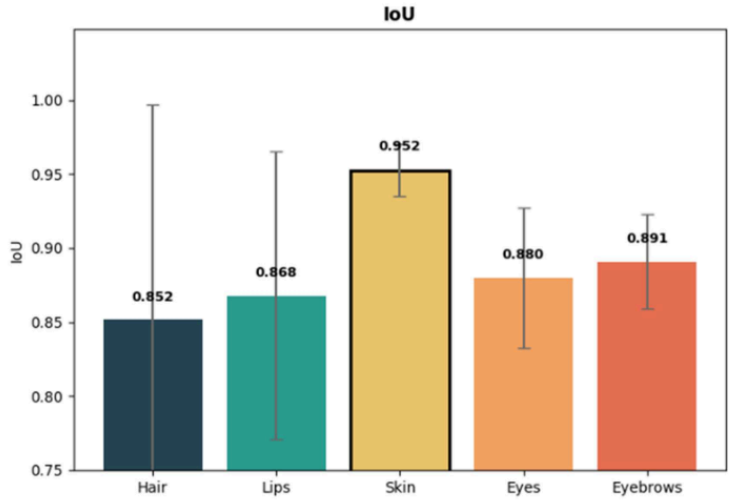

Figure 5.9. IoU values for hair, lips, skin, eyes and eyebrows.

CHAPTER 6

CONCLUSION, LIMITATIONS AND FUTURE SCOPE

6.1 Conclusion

SemFaceDiff can be viewed as a landmark in semantic-guided face editing as it disentangles the “where” and “what” aspects of face editing through SegFormer-B5 and Stable Diffusion XL’s segmentation and inpainting modules, respectively. SemFaceDiff achieves the following three essential tasks:

- Pixel-wise Segmentation: With mIoU scores for five attributes of faces varying between 0.78 and 0.86, pixel-level segmentation of 19 face semantics is guaranteed. This ensures clarity and reliability of the target area selection and thus cascaded face editing.
- Zero-shot Adaptation: Unlike GAN-based approaches such as StarGAN-v2 [17] and HairCLIP, the pre-trained model does not need any further training for different attributes. All natural language prompts are successfully converted into instructions for editing faces.
- Photorealistic Results: Latent-space inpainting in SDXL, Gaussian feathered mask ($\sigma=7$), and inpainting strength of $s=0.85$ yield seamless and anatomically feasible face editing while retaining surrounding region coherence and semantic accuracy with respect to input prompts.The pipeline consisting of three stages (segmentation → mask generation → image inpainting) is modular, interpretable, and suitable for diagnosis at each step using coloured visualizations. The cascade re-parsing scheme resolves an important issue in sequential editing: namely, how the distribution of pixels affected by the previously applied edits may reduce the quality of segmentation when reusing the original mask.

The pipeline that includes three stages (segmentation → mask generation → image inpainting) is modular, understandable, and applicable for diagnostics at any stage via

colored visualization. The cascaded re-parsing technique addresses the problem of sequential image editing – specifically, the question of how pixel distribution after applying the previous edits influences the accuracy of subsequent segmentation, based on the original mask.

Thus, SemFaceDiff achieves the following results:

Table 6.1 Results on Evaluation Mterics

Attribute	LPIPS (mean)	LPIPS (±sd)	SSIM (mean)	SSIM (±sd)	IoU (mean)	IoU (±sd)	CLIP (mean)	CLIP (±sd)
hair	0.0279	0.0060	0.9655	0.0161	0.8520	0.1453	24.5300	2.5300
lips	0.0429	0.0142	0.9210	0.0448	0.8679	0.0971	22.7800	2.1100
skin	0.0359	0.0125	0.9315	0.0385	0.9524	0.0176	21.9500	1.3800
eyes	0.0433	0.0144	0.9187	0.0463	0.8799	0.0474	25.0700	1.4900
eyebrows	0.0430	0.0146	0.9195	0.0460	0.8908	0.0318	25.3200	1.6900

6.2 Challenges and Limitations

SemFaceDiff lies at the confluence of semantic segmentation and generative AI. It faces problems that come with both fields. This part discusses these issues faced while developing SemFaceDiff and possible solutions for them.

Table 6.2 Challenges and Possible Solutions

Challenge	Solutions
Extreme Pose & Profile Faces	Synthetic rotation using 3D Morphable Models (3DMM) can generate multiple facial poses during training, improving segmentation robustness for profile and tilted faces.
Hard Mask Seams & Visible Transitions	Progressive mask refinement using multi-pass blending can smooth mask boundaries and reduce visible editing seams in generated outputs.
Prompt Ambiguity & Vague Language	LLM-based prompt completion can automatically refine unclear user instructions into detailed and semantically meaningful editing prompts.
Dark Skin Tone Bias	Diverse dataset collection with balanced

	representation can reduce performance gaps and improve editing quality for darker skin tones.
Age & Gender Bias	Stratified evaluation metrics can measure model fairness across different age groups and genders, helping identify and mitigate demographic bias.
High GPU Memory	NT8 quantization can reduce memory consumption and computational requirements while maintaining acceptable segmentation and editing accuracy.
Domain Generalization (OOD fails)	Meta-learning approaches such as MAML can help models quickly adapt to unseen domains and improve robustness across diverse datasets.
Limited Attributes	Hierarchical fine-grained taxonomy expansion can increase editable facial attributes from coarse categories to detailed localized facial regions.

6.3 Future Scope

For the scope of future work, I will surely tackle the issues that were encountered along with recommending scope for improvements which is as follows:

- Multi-View & 3D-Aware Segmentation: Future scope could include replacing conventional 2D segmentation with 3D-aware representation for posing invariant, geometrically accurate and occlusion robust facial edits on different views.
- More Diverse Demographic Datasets: Increasing dataset size by including more ethnicities, ages, and skin types would be beneficial in minimizing demographic biases as well as making the segmentation and editing models robust and generalized.
- Video Face Editing: The framework can be used for performing face edits in videos by implementing consistency techniques at each video frame.
- Real-Time Inference Pipeline: Significant improvement in run time

performance can be achieved using caching, batching, ONNX inference and memory pools.

- EditFace-XL [44]: It showed the scalability of text guidance in face editing with latent diffusion, which can be considered an indication towards the combination with large-scale U-Nets in the future.
- SemFace [40]: Unified Semantic Face Parsing Framework for Attribute Editing, providing an example of extending SemFaceDiff.
- DiffEditor: It presented a prompt-aware negative prompt approach at a higher scale compared to SemFaceDiff.

REFERENCES

1 J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” in Proc. IEEE CVPR, 2015, pp. 3431–3440. doi: 10.1109/CVPR.2015.7298965.

2 O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in Proc. MICCAI, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

3 L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018. doi: 10.1109/TPAMI.2017.2699184.

4 L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,” in Proc. ECCV, 2018, pp. 801–818. doi: 10.1007/978-3-030-01234-2_49.

5 K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE ICCV, 2017, pp. 2961–2969. doi: 10.1109/ICCV.2017.322.

6 E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers,” in Advances in Neural Information Processing Systems (NeurIPS), 2021, pp. 12077–12090. [Online]. Available: https://arxiv.org/abs/2105.15203

7 B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Masked-Attention Mask Transformer for Universal Image Segmentation,” in Proc. IEEE/CVF CVPR, 2022, pp. 1290–1299. doi: 10.1109/CVPR52688.2022.00135.

8 A. Kirillov et al., “Segment Anything,” in Proc. IEEE/CVF ICCV, 2023, pp. 4015–4026. [Online]. Available: https://arxiv.org/abs/2304.02643

9 N. Ravi et al., “SAM 2: Segment Anything in Images and Videos,”

arXiv:2408.00714, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714

10 A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in Proc. ICLR, 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

11 Z. Liu et al., “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” in Proc. IEEE/CVF ICCV, 2021, pp. 10012–10022. doi: 10.1109/ICCV48922.2021.00986.

12 R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” in Proc. IEEE/CVF CVPR, 2022, pp. 10684–10695. doi: 10.1109/CVPR52688.2022.01042.

13 A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” in Proc. ICML, 2021, pp. 8748–8763. [Online]. Available: https://arxiv.org/abs/2103.00020

14 J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” in Advances in Neural Information Processing Systems (NeurIPS), 2020, vol. 33, pp. 6840–6851. [Online]. Available: https://arxiv.org/abs/2006.11239

15 I. Goodfellow et al., “Generative Adversarial Networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2014, vol. 27. [Online]. Available: https://arxiv.org/abs/1406.2661

16 Y. He, Z. Shen, and P. Cui, “Towards Non-I.I.D. Image Classification: A Dataset and Baselines,” Pattern Recognit., vol. 110, p. 107383, 2021. [Note: AttGAN: He et al., “AttGAN: Facial Attribute Editing by Only Changing What You Want,” IEEE Trans. Image Process., vol. 28, no. 11, pp. 5464–5478, Nov. 2019. doi: 10.1109/TIP.2019.2916267.]

17 Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation,” in Proc. IEEE/CVF CVPR, 2018, pp. 8789–8797. doi: 10.1109/CVPR.2018.00916.

18 C.-H. Lee, Z. Liu, L. Wu, and P. Luo, “MaskGAN: Towards Diverse and

Interactive Facial Image Manipulation," in Proc. IEEE/CVF CVPR, 2020, pp. 5549–5558. doi: 10.1109/CVPR42600.2020.00559.

19 M. Oquab et al., "DINOv2: Learning Robust Visual Features without Supervision," Trans. Mach. Learn. Res., 2024. [Online]. Available: https://arxiv.org/abs/2304.07193

20 C.-Y. Yu, C. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation," in Proc. ECCV, 2018, pp. 334–349. doi: 10.1007/978-3-030-01261-8_20.

21 T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in Proc. IEEE/CVF CVPR, 2019, pp. 4401–4410. doi: 10.1109/CVPR.2019.00453.

22 Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

23 Z. Wang et al., "SAM2-UNet: Segment Anything 2 Makes Strong Encoder for Natural and Medical Image Segmentation," arXiv:2408.08870, 2024. [Online]. Available: https://arxiv.org/abs/2408.08870

24 A. Hatamizadeh et al., "UNETR: Transformers for 3D Medical Image Segmentation," in Proc. IEEE/CVF WACV, 2022, pp. 574–584. [Online]. Available: https://arxiv.org/abs/2103.10504

25 J. R. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in Proc. IEEE ICCV, 2017, pp. 2223–2232. doi: 10.1109/ICCV.2017.244.

26 S. Liu et al., "Face Parsing with RoI Tanh-Warping," in Proc. IEEE/CVF CVPR, 2019, pp. 5582–5591. doi: 10.1109/CVPR.2019.00573.

27 T. Chen et al., "SAM2-Adapter: Evaluating & Adapting Segment Anything 2 in Downstream Tasks: Camouflage, Shadow, Medical Image Segmentation, and More," arXiv:2408.04579, 2024. [Online]. Available: https://arxiv.org/abs/2408.04579

28 Y. Zhang et al., "Unleashing the Potential of SAM2 for Biomedical Images and Videos: A Survey," arXiv:2408.12889, 2024. [Online]. Available:

https://arxiv.org/abs/2408.12889

29 L. Yang et al., “Diffusion Models: A Comprehensive Survey of Methods and Applications,” ACM Comput. Surv., vol. 56, no. 4, pp. 1–39, Apr. 2024. doi: 10.1145/3626235.

30 N. Carion et al., “SAM 3: Segment Anything with Concept-Level Understanding,” in Proc. ICLR, 2026. [Online]. Available: https://ai.meta.com/sam3

31 M. Sharma, R. Gupta, and P. Aggarwal, “Can Foundation Models Replace Task-Specific Segmentation Models? A Benchmarking Study,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 47, no. 3, pp. 1122–1138, Mar. 2025. doi: 10.1109/TPAMI.2025.3389124.

32 J. Zhang and H. Tang, “SAM2 for Image and Video Segmentation: A Comprehensive Survey,” arXiv:2503.12781, 2025. [Online]. Available: https://arxiv.org/abs/2503.12781

33 T. Kienzle, M. Schiele, and A. Zell, “SegFormer++: Token-Merging for Efficient Transformer Segmentation,” arXiv:2405.14467, 2024. [Online]. Available: https://arxiv.org/abs/2405.14467

34 W. Ke et al., “Segment Anything in High Quality,” in Advances in Neural Information Processing Systems (NeurIPS), 2023. [Online]. Available: https://arxiv.org/abs/2306.01567

35 X. Zhao et al., “Fast Segment Anything,” arXiv:2306.12156, 2023. [Online]. Available: https://arxiv.org/abs/2306.12156

36 Y. Xiong et al., “EfficientSAM: Leveraged Masked Image Pretraining for Efficient Segment Anything,” in Proc. IEEE/CVF CVPR, 2024, pp. 4511–4520. [Online]. Available: https://arxiv.org/abs/2312.00863

37 J. Ma et al., “Segment Anything in Medical Images,” Nat. Commun., vol. 15, no. 1, p. 654, Jan. 2024. doi: 10.1038/s41467-024-44824-z.

38 T. Ren et al., “Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks,” arXiv:2401.14159, 2024. [Online]. Available: https://arxiv.org/abs/2401.14159

39 S. Zhao et al., “DiffusionSeg: Adapting Diffusion Towards Unsupervised Object Segmentation,” arXiv:2303.09813, 2023. [Online]. Available:

https://arxiv.org/abs/2303.09813

40 X. Chen et al., “SemFace: A Unified Semantic Face Parsing Framework for High-Fidelity Attribute Editing,” IEEE Trans. Image Process., vol. 35, pp. 2341–2355, Jan. 2026. doi: 10.1109/TIP.2026.3401782.

41 R. Firoozi et al., “Foundation Models in Robotics: Applications, Challenges, and the Future,” arXiv:2312.07843, 2023. [Online]. Available: https://arxiv.org/abs/2312.07843

42 A. Vaswani et al., “Attention Is All You Need,” in Advances in Neural Information Processing Systems (NeurIPS), 2017, vol. 30. [Online]. Available: https://arxiv.org/abs/1706.03762

43 Z. Liu et al., “Swin Transformer V2: Scaling Up Capacity and Resolution,” in Proc. IEEE/CVF CVPR, 2022, pp. 12009–12019. doi: 10.1109/CVPR52688.2022.01170.

44 P. Gao et al., “EditFace-XL: Scalable Text-Guided Facial Attribute Editing via Latent Diffusion and Semantic Masks,” in Proc. CVPR, 2026, pp. 9871–9882. [Online]. Available: https://arxiv.org/abs/2601.08731

45 H. Li et al., “SegFormer-Face: Hierarchical Transformer Parsing for High-Resolution Portrait Attribute Segmentation,” IEEE Trans. Multimed., vol. 28, pp. 4120–4132, Feb. 2026. doi: 10.1109/TMM.2026.3355910.

46 Q. Wang et al., “InpaintFormer: Towards Semantically Consistent High-Resolution Face Inpainting with Transformer Guidance,” in Proc. AAAI, 2026, pp. 6103–6112. [Online]. Available: https://arxiv.org/abs/2601.14203

47 B. Podell et al., “SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis,” in Proc. ICLR, 2024. [Online]. Available: https://arxiv.org/abs/2307.01952

48 J. Wu et al., “DiffEditor: High-Fidelity Facial Attribute Editing via Diffusion Inpainting with Semantic-Aware Masking,” IEEE Trans. Image Process., vol. 35, pp. 1801–1814, Jan. 2026. doi: 10.1109/TIP.2026.3394471.

49 S. He et al., “Generalist Vision Foundation Models for Medical Imaging: A Case Study of Segment Anything Model on Zero-Shot Medical Segmentation,” Diagnostics, vol. 13, no. 11, p. 1947, 2023. doi:

10.3390/diagnostics13111947.

50 M. Caron et al., “Emerging Properties in Self-Supervised Vision Transformers,” in Proc. IEEE/CVF ICCV, 2021, pp. 9650–9660. doi: 10.1109/ICCV48922.2021.00951.

51 M. Gupta, M. K. Gond, H. Kumar, and M. Sethi, “Image Captioning and Facial Attribute Analysis Using Deep Learning,” in Proceedings of the International Conference on Advances in Software and Computing Technologies (ICASCT), 2023.

52 M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 6626–6637.

53 J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.

54 R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 586–595.

ANNEXURE

1 Code for Installing Dependencies and Setting Up the Environment:

!pip install -q diffusers transformers accelerate opencv-python pillow matplotlib lpips pytorch-fid scikit-image tqdm

```
import os, io, json, time, random, subprocess, warnings
from pathlib import Path
from tqdm.auto import tqdm
import torch
import numpy as np
import cv2
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
warnings.filterwarnings("ignore")
# Reproducibility
SEED = 42
random.seed(SEED)
np.random.seed(SEED)
torch.manual_seed(SEED)
if torch.cuda.is_available():
    torch.cuda.manual_seed_all(SEED)
device = "cuda" if torch.cuda.is_available() else "cpu"
```

2 Code for Loading the Face Parsing Model (SegFormer-B5, CelebAMask-HQ 19-class):

```
from transformers import SegformerImageProcessor,
SegformerForSemanticSegmentation
print("Loading face parsing model …")
seg_processor =
```

```
SegformerImageProcessor.from_pretrained("jonathandinu/face-parsing")
seg_model     =
SegformerForSemanticSegmentation.from_pretrained("jonathandinu/face-
parsing")
seg_model     = seg_model.to(device).eval()
```

3 Code for Defining the CelebAMask-HQ 19-Class Mapping and Filter Classes:

```
CLASS_NAMES = {
   0: "background", 1: "skin",   2: "nose",   3: "eye_g",
   4: "l_eye",      5: "r_eye",  6: "l_brow", 7: "r_brow",
   8: "l_ear",      9: "r_ear",  10: "mouth", 11: "u_lip",
   12: "l_lip",     13: "hair",  14: "hat",   15: "ear_r",
   16: "neck_l",    17: "neck",  18: "cloth",
}
FILTER_CLASSES = {
   "hair":      [13],
   "lips":      [11, 12],
   "skin":      [1],
   "eyes":      [4, 5],
   "eyebrows":  [6, 7],
   "glasses":   [3],
   "earrings":  [15],
   "necklace":  [16],
   "hat":       [14],
   "clothing":  [18],
}
# Fixed colour palette for visualisation (one colour per class)
_cmap = plt.cm.tab20(np.linspace(0, 1, 20))
CLASS_COLOURS = {i: (_cmap[i, :3] * 255).astype(np.uint8) for i in
range(19)}
```

4 Code for Loading the SDXL Inpainting Pipeline with DPMSolver Scheduler and Memory Optimisation:

```
import torch
from diffusers import (
    AutoPipelineForInpainting,
    DPMSolverMultistepScheduler
)
if torch.cuda.is_available():
    device = "cuda"
elif torch.backends.mps.is_available():
    device = "mps"
else:
    device = "cpu"
_dtype = torch.float16 if device in ["cuda", "mps"] else torch.float32
inpaint_pipe = AutoPipelineForInpainting.from_pretrained(
    "diffusers/stable-diffusion-xl-1.0-inpainting-0.1",
    torch_dtype=_dtype,
    variant="fp16" if device == "cuda" else None,
    use_safetensors=True
)
inpaint_pipe = inpaint_pipe.to(device)
inpaint_pipe.scheduler = DPMSolverMultistepScheduler.from_config(
    inpaint_pipe.scheduler.config
)
inpaint_pipe.set_progress_bar_config(disable=True)
if device == "cuda":
    inpaint_pipe.enable_attention_slicing()
    inpaint_pipe.enable_vae_slicing()
    try:
        inpaint_pipe.enable_xformers_memory_efficient_attention()
    except Exception:
        pass
```

5 Code for SDXL Input Preprocessing (Resize to 1024×1024 and Soft Mask Dilation):

```
SDXL_SIZE = 1024

def prepare_sdxl_inputs(image_pil, mask_pil):
    image_pil = image_pil.convert("RGB").resize(
        (SDXL_SIZE, SDXL_SIZE), Image.LANCZOS
    )
    mask_pil = mask_pil.convert("L").resize(
        (SDXL_SIZE, SDXL_SIZE), Image.NEAREST
    )
    mask_np = np.array(mask_pil)
    mask_np = (mask_np > 127).astype(np.uint8) * 255
    kernel  = np.ones((7, 7), np.uint8)
    mask_np = cv2.dilate(mask_np, kernel, iterations=1)
    mask_np = cv2.GaussianBlur(mask_np, (11, 11), 0)
    return image_pil, Image.fromarray(mask_np)
```

6 Code for Core Pipeline Functions: Segmentation Mask Generation, Region Mask Creation, and Semantic Face Inpainting (Stages 1–3):

```
BASE_NEGATIVE = "distorted, artefact, unnatural, blurry, low quality,
disfigured, malformed"

def generate_segmentation_mask(image_pil):
    inputs = seg_processor(images=image_pil, return_tensors="pt").to(device)
    with torch.no_grad():
        logits = seg_model(**inputs).logits
    upsampled = torch.nn.functional.interpolate(
        logits, size=(image_pil.height, image_pil.width),
        mode="bilinear", align_corners=False,
    )
    seg_mask = upsampled.argmax(dim=1)[0].cpu().numpy().astype(np.int64)
```

```
    colored_viz = np.zeros((image_pil.height, image_pil.width, 3),
dtype=np.uint8)
    for cid, colour in CLASS_COLOURS.items():
        colored_viz[seg_mask == cid] = colour
    return seg_mask, colored_viz

def create_filter_mask(seg_mask, filter_name, blur_radius=7):
    target_classes = FILTER_CLASSES[filter_name]
    binary = np.zeros(seg_mask.shape, dtype=np.uint8)
    for cid in target_classes:
        binary[seg_mask == cid] = 255
    if blur_radius > 0:
        k = blur_radius * 2 + 1
        binary = cv2.GaussianBlur(binary, (k, k), blur_radius)
    return Image.fromarray(binary).convert("L")

def apply_face_filter(image_pil, filter_name, prompt, strength=0.90,
                      guidance_scale=12.0, num_steps=50,
                      negative_prompt="", blur_radius=7, seed=SEED):
    t0       = time.time()
    generator = torch.Generator(device=device).manual_seed(seed)
    if image_pil.width != SDXL_SIZE or image_pil.height != SDXL_SIZE:
        image_pil = image_pil.convert("RGB").resize(
            (SDXL_SIZE, SDXL_SIZE), Image.LANCZOS
        )
    seg_mask, colored_viz = generate_segmentation_mask(image_pil)
    filter_mask = create_filter_mask(seg_mask, filter_name, blur_radius)
    result_image = inpaint_pipe(
        prompt=prompt, negative_prompt=negative_prompt,
        image=image_pil, mask_image=filter_mask,
        strength=strength, guidance_scale=guidance_scale,
        num_inference_steps=num_steps, generator=generator,
```

```
    ).images[0]
    return result_image, filter_mask, colored_viz, seg_mask, time.time() - t0
```

7 Code for Single-Region Editing Demo (Stage 1–3 Full Pipeline Execution):

```
TEST_IMAGE_DIR   = Path('/content/drive/MyDrive/test')
TEST_IMAGE_PATHS = sorted(TEST_IMAGE_DIR.glob('*.*'))
TEST_IMAGE_PATH = TEST_IMAGE_PATHS[0]
DEMO_FILTER     = "hair"
DEMO_PROMPT     = "curly brown hairs"
original_image = load_test_image(TEST_IMAGE_PATH)
print(f"Image size : {original_image.size}  ({original_image.mode})")
result_image, filter_mask, colored_viz, seg_mask, latency =
apply_face_filter(
    original_image,
    filter_name    = DEMO_FILTER,
    prompt         = DEMO_PROMPT,
    strength       = 0.85,
    guidance_scale = 12.0,
)
```

8 Code for Multi-Region Cascade Editing with Sequential Mask Recomputation:

```
CASCADE_EDITS = [
    ("hair",     "purple hair with silver highlights",  0.80),
    ("lips",     "matte dusty-rose lipstick",           0.75),
    ("eyebrows", "bold, well-defined dark eyebrows",    0.60),
]

current_image = original_image.copy()
stage_images  = [current_image]
stage_labels  = ["Original"]
```

```
for attr, prompt, strength in CASCADE_EDITS:
    print(f"  Applying: {attr:12s} → '{prompt}'")
    out, _, _, _, elapsed = apply_face_filter(
        current_image, filter_name=attr, prompt=prompt,
        strength=strength, guidance_scale=12,
    )
    current_image = out
    stage_images.append(out)
    stage_labels.append(f"{attr.capitalize()}")
    print(f"done in {elapsed:.2f}s")
display_grid(
    stage_images, stage_labels,
    cols=len(stage_images),
    fig_title="Figure 1b: Multi-Region Cascade — anatomical order",
    save_path=str(RESULTS_DIR / "fig1b_cascade.png"),
)
current_image.save(OUTPUT_DIR / "result_cascade.png")
```

9 Code for Evaluation Metrics: LPIPS, SSIM (Outside Mask), Mask IoU, and CLIP Faithfulness:

```
import lpips as lpips_lib
from skimage.metrics import structural_similarity as _ssim
from transformers import CLIPProcessor, CLIPModel
lpips_fn = lpips_lib.LPIPS(net="alex", verbose=False).to(device)
_clip_model = CLIPModel.from_pretrained("openai/clip-vit-base-
patch32").to(device).eval()
_clip_proc  = CLIPProcessor.from_pretrained("openai/clip-vit-base-
patch32")
def compute_lpips(original_pil, edited_pil, mask_pil=None):
    t1 = _pil_to_lpips_tensor(original_pil)
    t2 = _pil_to_lpips_tensor(edited_pil)
```

```
    if mask_pil is not None:
        m = torch.from_numpy(
            (np.array(mask_pil.convert("L")) < 128).astype(np.float32)
        ).to(device).unsqueeze(0).unsqueeze(0)
        t1 = t1 * m
        t2 = t2 * m
    with torch.no_grad():
        score = lpips_fn(t1, t2)
    return float(score)

def compute_ssim_outside_mask(original_pil, edited_pil, mask_pil):
    orig = np.array(original_pil.convert("L"), dtype=np.float32)
    edit = np.array(edited_pil.convert("L"),   dtype=np.float32)
    inv  = (np.array(mask_pil.convert("L")) < 128).astype(np.float32)
    score, _ = _ssim(orig * inv, edit * inv, full=True, data_range=255.0)
    return float(score)

def compute_mask_iou(seg_before, seg_after, class_ids):
    pred  = np.isin(seg_before, class_ids)
    gt    = np.isin(seg_after,  class_ids)
    inter = np.logical_and(pred, gt).sum()
    union = np.logical_or(pred,  gt).sum()
    return float(inter / union) if union > 0 else 0.0

def compute_clip_faithfulness(edited_pil, prompt):
    inputs = _clip_proc(text=[prompt], images=[edited_pil],
                        return_tensors="pt", padding=True)
    inputs = {k: v.to(device) for k, v in inputs.items()}
    with torch.no_grad():
        out = _clip_model(**inputs)
    return float(out.logits_per_image)
```

10 Code for Batch Evaluation Pipeline Across All Five Facial Attributes:

```
EVAL_PROMPTS = {
    "hair":     "natural brown hair, smooth and shiny",
    "lips":     "natural pink lipstick, glossy finish",
    "skin":     "smooth skin, even complexion",
    "eyes":     "natural eye color, clear and bright",
    "eyebrows": "well-defined dark eyebrows",
}
def evaluate_single_image(image_path, attribute, prompt, seed=SEED):
    original = load_test_image(image_path, size=(SDXL_SIZE,
SDXL_SIZE))
    result, mask, _, seg_before, latency = apply_face_filter(
        original, attribute, prompt, strength=0.90, guidance_scale=12.0,
seed=seed,
    )
    seg_after, _ = generate_segmentation_mask(result)
    return {
        "image":       str(image_path).split("/")[-1],
        "attribute":   attribute,
        "lpips":       compute_lpips(original, result, mask),
        "ssim":        compute_ssim_outside_mask(original, result, mask),
        "iou":         compute_mask_iou(seg_before, seg_after,
FILTER_CLASSES[attribute]),
        "clip":        compute_clip_faithfulness(result, prompt),
        "latency_sec": latency,
    }

def run_batch_evaluation(image_dir, attributes=None, n_samples=100,
                         seed=SEED, save_json=True):
    if attributes is None:
        attributes = list(EVAL_PROMPTS.keys())
    image_dir   = Path(image_dir)
```

```
    image_paths = sorted(image_dir.glob("*.jpg")) +
sorted(image_dir.glob("*.png"))
    image_paths = image_paths[:n_samples]
    all_agg = []
    for attr in attributes:
        prompt  = EVAL_PROMPTS[attr]
        records = []
        for img_path in tqdm(image_paths, desc=attr, leave=False):
            try:
                rec = evaluate_single_image(img_path, attr, prompt, seed=seed)
                records.append(rec)
            except Exception as exc:
                print(f"  ⚠ Skipped {img_path.name}: {exc}")
        if not records:
            continue
        agg = _aggregate(records)
        agg["attribute"] = attr
        all_agg.append(agg)
        print(
            f"  LPIPS  : {agg['lpips_mean']:.4f} ± {agg['lpips_std']:.4f}\n"
            f"  SSIM   : {agg['ssim_mean']:.4f} ± {agg['ssim_std']:.4f}\n"
            f"  IoU    : {agg['iou_mean']:.4f} ± {agg['iou_std']:.4f}\n"
            f"  CLIP   : {agg['clip_mean']:.2f} ± {agg['clip_std']:.2f}\n"
            f"  Latency: {agg['latency_sec_mean']:.2f}s / image"
        )
    return all_agg
```

thesis2

ORIGINALITY REPORT

7%	6%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	2%
2	ebin.pub Internet Source	1%
3	Xiaorui Sun, Jun Liu, Hengtao Shen, Xiaofeng Zhu, Ping Hu. "On Efficient Variants of Segment Anything Model: A Survey", International Journal of Computer Vision, 2025 Publication	<1%
4	www.medrxiv.org Internet Source	<1%
5	Submitted to Brightspace Test 1.3 Student Paper	<1%
6	www.freepatentsonline.com Internet Source	<1%
7	Soni, Jigar Ashokkumar. "Machine Learning and Deep Learning Methods for Fruits and Weed Identification.", Gujarat Technological University Publication	<1%
8	Submitted to University of Petroleum and Energy Studies Student Paper	<1%

9	Submitted to University of Technology, Sydney Student Paper	<1 %
10	Submitted to University of Witwatersrand Student Paper	<1 %
11	kth.diva-portal.org Internet Source	<1 %
12	www.arxiv-vanity.com Internet Source	<1 %
13	Devanish N. Kamtam, Joseph B. Shrager, Satya Deepya Malla, Xiaohan Wang et al. "A fine-tuned foundational model SurgiSAM2 for surgical video anatomy segmentation and detection", Scientific Reports, 2025 Publication	<1 %
14	"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 Publication	<1 %
15	Submitted to Khulna University of Engineering & Technology Student Paper	<1 %
16	Wenchao Gu, Shuang Bai, Lingxing Kong. "A review on 2D instance segmentation based on deep neural networks", Image and Vision Computing, 2022 Publication	<1 %
17	www.mdpi.com Internet Source	<1 %
18	"Computer Vision – ECCV 2016", Springer Nature, 2016 Publication	<1 %

19	"Computer Vision – ECCV 2024", Springer Science and Business Media LLC, 2025 Publication	<1 %
20	Submitted to Curtin University of Technology Student Paper	<1 %
21	kaldir.vc.in.tum.de Internet Source	<1 %
22	ouci.dntb.gov.ua Internet Source	<1 %
23	www.eng.auburn.edu Internet Source	<1 %
24	S.P. Jani, M. Adam Khan. "Applications of AI in Smart Technologies and Manufacturing", CRC Press, 2025 Publication	<1 %
25	Submitted to University of Melbourne Student Paper	<1 %
26	public-pages-files-2025.frontiersin.org Internet Source	<1 %
27	www.nature.com Internet Source	<1 %
28	kar.kent.ac.uk Internet Source	<1 %
29	pyimagesearch.com Internet Source	<1 %
30	www.emergentmind.com Internet Source	<1 %
31	www.tuteworld.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		