ByeBye: A Zero-Shot Human Removal and Replacement Pipeline with Stylized Character Insertion

Ashwin Mahendran
Stanford Unviersity
mashwin@stanford.edu

Caleb Youngjae Whang Choe
Stanford Unviersity
choe27@stanford.edu

Arihan Varanasi
Stanford Unviersity
arihan@stanford.edu

Abstract

We present **ByeBye**, a modular zero-shot pipeline for removing humans from videos and replacing them with stylized characters. Our method integrates YOLOv8 for detection, SAM for segmentation, LaMa for high-fidelity inpainting, and Stable Diffusion with ControlNet for pose-aware character generation. This frame-by-frame pipeline enables rapid experimentation using powerful off-the-shelf models. Through detailed ablation studies, we find that the combination of YOLOv8s, SAM-ViT-B, and **LaMa Dilated** provides the best balance between perceptual quality and computational efficiency. To mitigate temporal inconsistencies inherent in frame-by-frame generation, we conduct a targeted hyperparameter sweep. We determine that a **CFG guidance scale of 9** and a **strength value of 0.7** result in the lowest mean and variance of LPIPS, indicating the highest perceptual coherence across frames. Additionally, we incorporate LoRA models fine-tuned for specific characters, which further enhance temporal consistency—particularly in preserving clothing and background elements—despite introducing minor degradations in facial fidelity. Our results demonstrate that a thoughtfully configured modular pipeline can achieve high-quality human removal and stylized character replacement in video without requiring custom training. This work lays a foundation for future extensions, including multi-human tracking, pose-aware conditioning, and temporally coherent video diffusion models.

1. Introduction

With the increasing use of video surveillance in both public and private environments, concerns over privacy are growing rapidly. One promising solution is video inpainting: the process of removing selected regions (such as people) from video frames and filling in the background so that the video remains visually coherent. While originally developed for restoration and anonymization, this technique also opens the door to creative inpainting, where masked regions can be filled with stylized patterns, textures, or entirely new characters.

In this project, we develop a modular pipeline for human-aware video inpainting, where our pipeline detects, segments, and removes humans frame by frame in a video sequence and fills the masked region with the background. Our background inpainted output appears clean and temporally consistent enabling creative addons. In particular, we go beyond just removal and explore cartoon/anime character replacement. Here we generate a new character which mimics the pose of the removed human which produces a fun, visually-appealing video. This task combines both generative models, pose estimation as well as background inpainting. Our pipeline is a one-shot approach leveraging powerful off-the-shelf models into a novel configuration that enables for quick swapping and experimentation.

The input into our system is a video containing a single human which is then decomposed into RGB frames. Each frame is first passed into a **YOLOv8** [7] model which generates bounding boxes around the human. These boxes are then passed into the Segment Anything Model **(SAM)** [8] to generate segmentation masks on top of the human. For the task of background inpainting we utilize the state-of-the-art **LaMa model** [14]. Additionally, as a baseline for background inpainting we experimented with the available **OpenCV based blur** inpainting which demonstrated less coherent inpainting. The output here is a folder of frames containing the inpainted background. For the stylized character replacement, we extract the 2D pose of the human for each frame using **OpenPose** [2]. Given the pose we use a **Stable Diffusion** [13] augmented with **ControlNet** [18] to generate anime characters that replicate the original human's pose and orientation. We also add a **LoRA** [6] and experiment with initialization techniques to achieve greater temporal inconsistencies in the generated video. The output here is a set of frames of the newly drawn character. We then overlay these frames using the SAM masks onto the inpainted background to see the new character on the original background.

This modular pipeline enables us to experiment with various combinations of different YOLO models for detection, different mask generators (different lightweight and heavy versions of SAM), and inpainting models (various versions of LaMa along with OpenCV blur). We perform an ablation study to evaluate how switching out different components affects the final output in terms of both perceptual quality and computational cost. Interestingly, we are able to find out that lightweight YOLO and SAM models along with the base LaMa model is able to achieve strong results comparable to that of the stronger models while still taking significantly less performance costs and time. Additionally, we can see how stylization is able to improve in both quality and temporal consistency when a LoRA is applied on top of the stable diffusion.

2. Related Work

The fields of video inpainting and style transfer have many established subfields in computer vision such as object detection, segmentation, image inpainting, pose estimation, and text based image generation. Each of these areas have existing systems in isolation but few systems tackle the full pipeline end to end. Our work focuses on merging these models into a modular, zero-shot pipeline enabling background inpainting and character insertion.

Object detection remains a cornerstone of vision systems, with detectors such as YOLOv3 [11] and YOLOv8 [7] offering speed and accuracy suitable for live object detection. In our pipeline, YOLOv8 serves as the first step in locating humans. The task of segmentation has evolved from earlier models such as DeepLab [3] which utilizes dilated convolutions and a CNN network to assign class labels to every pixel and Mask R-CNN [5] which is an extension to the Faster R-CNN [12] that adds a prediction branch for pixel level segmentation. These approaches built the foundation for Meta's Segment Anything Model (SAM) [8] capable of producing high quality segmentation masks.

There have been a variety of image inpainting techniques. Including the Telea's algorithm [15] which is acting as our baseline in the OpenCV default inpaint. We are using LaMa [14] as our state of the art approach for inpainting which works based off of frequency-aware convolutions and large receptive fields to generate globally coherent image completions.

For pose-guided generation, OpenPose [2] is a widely used tool for extracting 2D key points from humans in motion, and it serves as the extraction in our character replacement pipeline. Other existing methods include Mediapipe [9] created for pose extraction. To generate the new characters we are using Stable Diffusion [13] for prompt based image generation. We build upon this by using ControlNet [18] which forces Stable Diffusion to generate images using pose as input.

Recent papers like AnimateDiff [4] attempt to generate entire videos using end-to-end video diffusion models. AnimateDiff works great at producing temporally consistent videos; however, it lacks support for video generation based on pose-guidance. While very few extensions exist, they have non-trivial setup that are more computationally heavy as well as less modular.

In contrast, our work presents a fully modular, frame-by-frame pipeline that supports both human removal and stylized character insertion without the need for any training or fine-tuning. Each component of detection, segmentation, inpainting, pose estimation, and stylized generation can be independently swapped or ablated. This flexibility allows us to evaluate a broad range of combinations and draw insights into how design choices impact perceptual quality, temporal consistency, and computational cost. Our approach demonstrates that a thoughtfully composed zero-shot system using off-the-shelf tools can produce interesting results, while being easier to use, adapt, and extend.

3. Dataset and Features

We evaluate our pipeline on videos from the DAVIS 2017 dataset [10], a standard benchmark in video object segmentation. We select a subset of videos that feature a single prominent human subject, allowing for clean pose extraction, segmentation, and stylization. This dataset provides diverse scenes with consistent camera motion, making it well-suited for evaluating inpainting and frame-wise video generation tasks.

While we use DAVIS because of its easy availability, our pipeline can work on any video with a single human. Currently our pipeline is optimized for inpainting and replacing a single human, but it can be easily adapted to multiple human scenes by using TrackAnything [17] to track single or multiple people to replace. Frames are extracted from each video and passed through the pipeline independently.

After passing in our image frames into SAM we save all the binary segmentation masks and use them as input to LaMa for inpainting. Additionally, when calculating the pose with OpenPose and before passing into the character generator pipeline (Stable DIffusion, ControlNet, and LoRA) we resize all the frames to by 512 by 512 pixels as these models expect. We then reshape them back into their original shape when pasting them back into the original video.

4. Methods

In this section, we describe the architecture of our modular pipeline for one-shot video inpainting and character replacement pipelines (see Figure 1). Our system works on a frame by frame basis while also allowing us to swap out different models at each stage without requiring time-intensive retraining. We experiment with various different YOLO models, SAM models, LaMa models, as well as the default OpenCV inpaint. We evaluated each component in

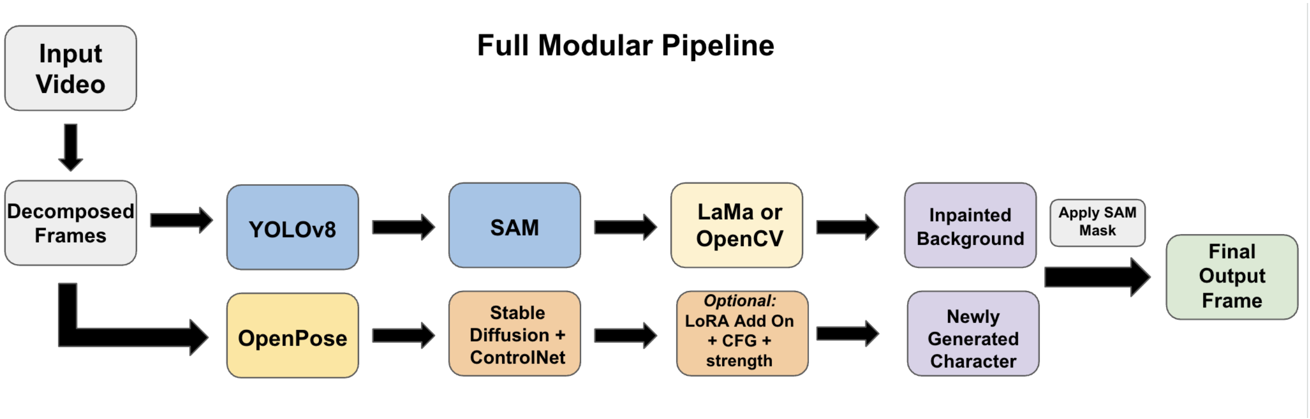


Figure 1. Modular pipeline for human-aware video inpainting and stylized character replacement. Each component: detection, segmentation, inpainting, and generation can be independently ablated or swapped.

an ablation study according to various different quantitative inpainting metrics. Additionally, we also explored the compute time for various combinations of larger parameter versus simpler models in order to determine what pipeline is able to provide the best results for fast, low-compute, results.

For the character replacement task we first extracted pose using OpenPose and experimented with stable diffusion plus controlnet add on to generate characters in the same pose as the original video. We can then overlay these generated videos with the earlier SAM masks to make the final video. Optional add-ons to this pipeline for better character generation and temporal consistency is using a LoRA model for a specific character and changing the initialization strength of the diffusion model.

4.1. Inpainting Task

Here our goal is to remove the human from each frame and inpaint the masked region with a high-quality representation of the background. This is achieved with three steps, **first** Human Detection, **second** Segmentation, and **third** inpainting.

4.1.1 Human Detection (YOLOv8)

We utilize the state-of-the-art object detection model **YOLOv8** (You Only Look Once) to extract bounding boxes around humans in each video frame. YOLO is a single-stage object detector that performs classification and localization in a single forward pass of a convolutional neural network, making it highly efficient for real-time use. Lighter variants, such as YOLOv8s, are particularly effective in scenarios like ours, where each video frame must be processed independently and quickly.

YOLOv8 processes an input RGB image of shape $\mathbb{R}^{H\times W\times 3}$ and outputs a tensor containing bounding box predictions, each represented as:

$$(x, y, w, h, c, p_1, \ldots, p_n)$$

where (x, y) are the center coordinates of the box, (w, h) are its dimensions, c is the objectness confidence, and p_i are class probabilities. YOLO's architecture predicts these values over a grid applied to the input feature map, and is trained using a multi-part loss that includes bounding box regression, object confidence, and classification terms.

In our modular pipeline, each video is decomposed into frames, which are passed into YOLOv8 to detect human bounding boxes. These boxes are critical for generating pixel-accurate masks with SAM in the next stage of the pipeline. We conduct an ablation study using multiple YOLOv8 variants—`YOLOv8s`, `YOLOv8m`, `YOLOv8l`, and `YOLOv8x`—to evaluate how model size affects both accuracy and runtime.

As shown in Table 1, while YOLOv8x yields the most accurate detections, it is also the slowest. We found that YOLOv8s offers a strong trade-off, producing high-quality bounding boxes for our single-human scenes at significantly lower computational cost, making it the best fit for our one-shot inpainting pipeline.

Model	Size (Millions of Parameters)	Speed (FPS)
YOLOv8s	11.2	144
YOLOv8m	25.9	96
YOLOv8l	43.7	70
YOLOv8x	68.2	53

Table 1. Comparison of YOLOv8 variants. YOLOv8s is the lightest and fastest, suitable for real-time usage. YOLOv8m offers a trade-off between speed and accuracy, while YOLOv8l and YOLOv8x are more accurate but increasingly compute-intensive. Speed and parameter size are from Ultralytics benchmarks [7].

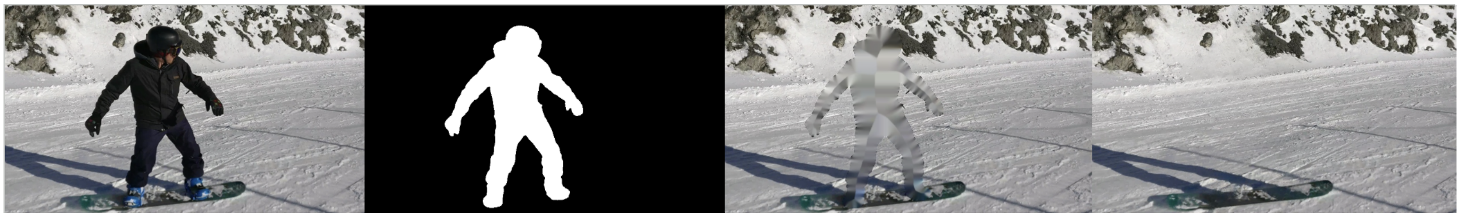


Figure 2. On the far left we have the input image, second we have the segmentation mask generated by SAM-H, third we have the background inpainting provided by OpenCV, and on the far right we have the inpainted background generated by LaMa MultiPass.

4.1.2 Segmentation and Mask Creation

To extract high-quality human masks from YOLOv8's predicted bounding boxes, we use the Segment Anything Model (SAM), a general-purpose segmentation framework developed by Meta AI. SAM is capable of producing accurate masks given prompts, including bounding boxes, points, or previous masks.

SAM has a transformer-based encoder-decoder architecture using a series of Vision Transformers. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$ and a prompt P (in our case, a bounding box), the model generates a binary segmentation mask $M \in \{0, 1\}^{H \times W}$, where 1 indicates pixels corresponding to the object of interest (or in our case human).

In our pipeline, we apply SAM to each frame using human bounding boxes predicted by YOLOv8. The output masks are used for two key purposes: first, to remove the human subject via background inpainting, and second, to composite the stylized character back into the original background.

We perform another ablation study using three SAM variants—`SAM-ViT-B`, `SAM-ViT-L`, and `SAM-ViT-H`—which differ in parameter size, input resolution, and runtime efficiency. As shown in Table 4.1.2, `SAM-ViT-H` yields the highest quality masks, wrapped close to the human figure.. However, `SAM-ViT-B` offers faster inference while producing masks that are sufficiently accurate for single-human scenes depending on the complexity of the video. This flexibility allows us to balance segmentation quality with computational cost, depending on a certain task's requirements.

Model	Backbone	Params (B)	Resolution
SAM-ViT-B	ViT-B	0.91	256×256
SAM-ViT-L	ViT-L	1.0	512×512
SAM-ViT-H	ViT-H	2.4	1024×1024

Table 2. Comparison of SAM variants used for human segmentation. SAM-ViT-B is a lightweight baseline; SAM-ViT-L balances accuracy and speed; SAM-ViT-H produces the highest-quality masks but is resource-intensive. Input resolutions correspond to the patch sizes expected by each ViT backbone. Values adapted from Kirillov et al. [8].

4.1.3 OpenCV Telea Inpainting

As a baseline for background inpainting, we first tried OpenCV's built-in implementation of Telea's algorithm. This method is fast and parameter-free, making it suitable for quick comparisons. However, it often struggles with complex backgrounds, motion edges, or semantic consistency.

Telea's algorithm fills in missing regions with surrounding pixel information. It is a fast marching method that propagates known pixel information from the boundary inward. At each step, the algorithm chooses the pixel with the smallest distance from the mask edge and estimates its value based on a weighted average of its known neighbors, with weights favoring directionality, smoothness, and proximity. Given a binary mask $M(x, y)$ and an input image $I(x, y)$, the algorithm fills $M = 1$ by solving:

$$I(x, y) = \frac{\sum_{(i,j) \in \mathcal{N}(x,y)} w_{i,j} \cdot I(i, j)}{\sum w_{i,j}}$$

where $\mathcal{N}(x, y)$ is the neighborhood of known pixels and $w_{i,j}$ is a weighting function that incorporates distance and image gradients.

This method is limited to local texture propagation and lacks any global scene understanding of the image. As shown in Figure 2, the inpainting result can appear smeared or blocky, clearly revealing the outline of the human, especially when large human masks cover complex background regions. Hence, more advanced models like LaMa must be used for higher quality inpainting.

4.1.4 LaMa Inpainting

We use LaMa (Large Mask Inpainting) as our core background restoration model. Unlike traditional pixel-based methods, LaMa is a learning-based approach that uses a Fast Fourier Convolution network built on top of a ResNet model to better capture global context. This enables the model to inpaint large missing regions with more coherent structure and texture.

LaMa encodes the masked image and learns to hallucinate realistic content in missing areas by leveraging both local and global receptive fields. It is trained using a combination of perceptual and adversarial losses to generate natural-looking results.

We experiment with three different LaMa variants to evaluate how architectural changes affect inpainting quality. First **LaMa Dilated** utilizes dilated convolutions which enables for faster inference but not as accurate as other LaMa models. **LaMa Aggressive** is more creative with its inpainting and more aggressively hallucinates the missing regions with more bold textures. **LaMa Multi-Pass** performs the inpainting model multiple times iteratively producing the best results at the greatest computational cost

Among the variants, LaMa Multi-Pass consistently delivered the best qualitative performance across diverse video scenes. However, even the lighter LaMa models outperformed the OpenCV Telea baseline by a wide margin. See Figure 2 for Multi-Pass LaMa background inpainting.

4.2. Stylized Character Replacement

4.2.1 Pose Extraction

To extract human poses from each frame of the input video, we use the OpenPose-based pose conditioning model from this ControlNet implementation [19] which is built on top of the original OpenPose framework. This model produces rendered pose images with colored joints and limb connections, which serve as the control input for the stable diffusion model.

This ControlNet-compatible version of OpenPose generates full-body skeleton visualizations directly as RGB images. We extract these poses from the original video frames before human removal and resize them to 512 by 512 (matching the diffusion model) to have a video of all the correct poses. These pose images are then used as the input for the ControlNet and Stable Diffusion pipeline for consistent, pose-matched characters. The image below shows a cropped version of the input/output of OpenPose.

4.2.2 Generative Model Pipeline

Our character generation module builds upon Stable Diffusion, a latent diffusion model (LDM) that synthesizes high-quality images from textual prompts. We use the anime tuned weights *stablediffusionapi/anything-v5* from [1], where we input both a positive prompt containing information like the character name, full-body, high quality and a negative prompt to suppress undesired features such as blurry, low-res, extra limbs, bad face (all aspects that stable diffusion struggle with). The model operates in the latent space $\mathcal{Z}$, where a noisy latent $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ is iteratively denoised over T steps to yield $\mathbf{z}_0$, the final latent representation. This process is governed by a trained denoising model ϵ_θ such that:

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}) \right) + \sigma_t \mathbf{n},$$

where α_t are predefined noise scheduling parameters, $\bar{\alpha}_t$ their cumulative product, $\mathbf{n} \sim \mathcal{N}(0, \mathbf{I})$, and $\mathbf{c}$ represents the conditional embeddings from the text prompt.

To guide the generation toward the structure of the removed human, we incorporate ControlNet conditioned on OpenPose skeletons. ControlNet anchors the generation by injecting structural guidance (like poses) directly into the diffusion layers, aligning outputs with spatial constraints.

4.2.3 Configurations and Tuning

To enhance stylistic control and improve temporal consistency in our generative pipeline, we incorporate three key components: Low-Rank Adaptation (LoRA), the `strength` parameter, and classifier-free guidance (CFG) scale. These allow us to fine-tune the balance between prompt adherence, pose conditioning, and creative flexibility.

LoRA: Low-Rank Adaptation. LoRA enables us to inject new concepts such as better performance on specific anime characters into a pretrained diffusion model by learning small, task-specific weight updates. Rather than retraining the entire model, LoRA freezes the original weights $W_0 \in \mathbb{R}^{d \times k}$ and learns two smaller matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$ such that:

$$W = W_0 + \alpha \cdot AB,$$

where $r \ll \min(d, k)$ and α is a scaling factor. These updates are applied to the attention layers of the U-Net backbone in Stable Diffusion, allowing for fast and memory-efficient injection of style-specific features. We are taking anime LoRA models sampled from here [16].

Strength. The `strength` parameter governs how closely the generation follows the conditioning input (i.e., the pose from ControlNet). Conceptually, it determines the blend between noise and the initial latent derived from the input image or condition:

$$z_0 = (1 - \texttt{strength}) \cdot z_{\text{input}} + \texttt{strength} \cdot z_{\text{noise}}.$$

Higher values push the model to conform more strictly to the pose, while lower values allow for greater generative freedom.

CFG Scale. Classifier-Free Guidance (CFG) controls how strongly the model follows the text prompt. During each denoising step, the noise prediction is modified using both conditioned and unconditioned predictions:

$$\epsilon = \epsilon_{\text{uncond}} + s \cdot (\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}),$$

where s is the guidance scale. A higher CFG scale enforces stronger alignment with the prompt, which in our case includes the anime character's name and visual description.

These hyperparameters together provide flexible control over the generative process, allowing us to balance temporal stability with prompt fidelity and stylistic realism.

4.3. Evaluation Metrics

To evaluate the performance of our modular pipeline across both the inpainting and character replacement tasks, we adopt a combination of perceptual and structural metrics. These include **Structural Similarity Index (SSIM)**, **Learned Perceptual Image Patch Similarity (LPIPS)**, and a temporal consistency measure over adjacent frames.

Structural Similarity Index (SSIM). SSIM evaluates structural and perceptual similarity by comparing luminance, contrast, and structure between images:

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

where μ_x, μ_y are the mean intensities, σ_x^2, σ_y^2 the variances, and σ_{xy} the covariance of the reference and generated images. C_1 and C_2 are constants to stabilize the division.

Learned Perceptual Image Patch Similarity (LPIPS). LPIPS is a deep-learning-based metric that evaluates perceptual similarity using feature maps from pretrained networks:

$$\text{LPIPS}(x, y) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (f_l^x - f_l^y)\|_2^2$$

where f_l^x and f_l^y are the features from layer l for images x and y, and w_l are learned weights. The final LPIPS score is the average over all feature layers.

Temporal Consistency. For evaluating temporal smoothness in videos, particularly across character replacement frames, we compute frame-to-frame LPIPS:

$$\text{LPIPS}_{\text{temporal}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \text{LPIPS}(f_t, f_{t+1})$$

where T is the number of frames and f_t denotes frame t. Lower LPIPS indicates more visually consistent outputs over time.

5. Experiments/Results/Discussion

5.1. Quantitative Inpainting Results

To understand the impact of mask generation on basic inpainting, we present two ablation tables for the *tennis* video: Table 3 for OpenCV Telea each with SSIM, LPIPS, and time per frame.

Mask Config + Telea	**SSIM ↑**	**LPIPS ↓**	**Time (ms/frame)**
YOLOv8s + SAM-ViT-B	0.964	0.057	46.2
YOLOv8m + SAM-ViT-B	0.964	0.057	45.7
YOLOv8l + SAM-ViT-L	0.963	0.059	45.6
YOLOv8x + SAM-ViT-H	0.963	0.058	45.7

Table 3. OpenCV Telea inpainting ablation for the *tennis* video with various mask configurations.

5.2. LaMa Technique Ablation

Table 4 presents SSIM, LPIPS, and per-frame inpainting time for the *tennis* video across different YOLO+SAM mask configurations and LaMa variants.

Mask Config + Variant	SSIM ↑	LPIPS ↓	Time (ms/frame)
YOLOv8s+SAM-ViT-B + Dilated	0.954	0.079	182
YOLOv8m+SAM-ViT-B + Dilated	0.954	0.079	183
YOLOv8l+SAM-ViT-L + Dilated	0.954	0.078	206
YOLOv8x+SAM-ViT-H + Dilated	0.954	0.078	216
YOLOv8s+SAM-ViT-B + Multi–Pass	0.950	0.080	550
YOLOv8m+SAM-ViT-B + Multi–Pass	0.950	0.080	522
YOLOv8l+SAM-ViT-L + Multi–Pass	0.950	0.080	520
YOLOv8x+SAM-ViT-H + Multi–Pass	0.950	0.080	537
YOLOv8s+SAM-ViT-B + Aggressive	0.943	0.084	886
YOLOv8m+SAM-ViT-B + Aggressive	0.943	0.084	922
YOLOv8l+SAM-ViT-L + Aggressive	0.943	0.084	859
YOLOv8x+SAM-ViT-H + Aggressive	0.943	0.083	999

Table 4. LaMa inpainting ablation for *tennis*: YOLO+SAM mask configurations and model variants with SSIM/LPIPS and compute time.

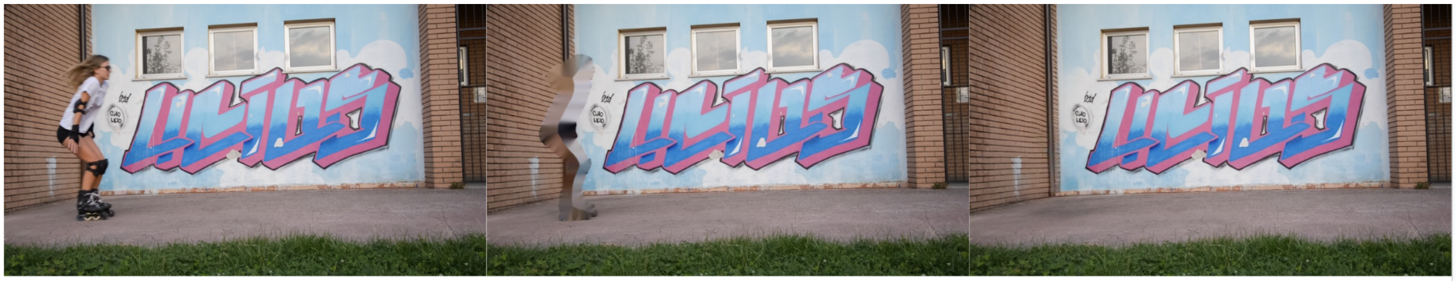


Figure 3. Original vs. OpenCV Telea vs. LaMa Multi–Pass for frame #0 of *roller blading*.

5.2.1 Average Across All Videos

Table 5 summarizes mean metrics over all sequences, where we also observe that the small mask configuration (YOLOv8s+SAM-ViT-B) consistently yields the best trade-off across both OpenCV and LaMa variants.

Method	SSIM ↑	LPIPS ↓	Time (ms/frame)
OpenCV Telea (YOLOv8s+SAM-ViT-B)	0.964	0.057	46.2
OpenCV NS (YOLOv8s+SAM-ViT-B)	0.965	0.052	54.4
LaMa Dilated (YOLOv8s+SAM-ViT-B)	0.954	0.079	182.1
LaMa Multi–Pass(YOLOv8s+SAM-ViT-B)	0.950	0.080	549.7
LaMa Aggressive (YOLOv8s+SAM-ViT-B)	0.943	0.084	886.5

Table 5. Average inpainting metrics and compute times across all 25 sequences.

5.3. Qualitative Comparison of a Representative Frame

Figure 3 shows frame #0 of the *roller blading* video: original, OpenCV Telea (YOLOv8s+SAM-ViT-B mask), and LaMa Multi–Pass (best variant). Note the superior preservation of the background building and graffiti by LaMa.

5.4. Discussion

Our systematic ablations reveal clear trade-offs across configurations. OpenCV Telea with YOLOv8s+SAM-ViT-B masks consistently achieves higher SSIM and lower LPIPS scores compared to LaMa Dilated, while also offering faster performance (46 vs. 182 ms/frame). This is because Telea's diffusion-based algorithm focuses on local pixel averaging, yielding smoother textures that quantitative metrics tend to reward. In contrast, LaMa's learning-based inpainting produces semantically informed details and structural coherence, which visual inspection shows are more realistic but can introduce perceptual differences that SSIM and LPIPS may penalize.

In LaMa experiments, Dilated produced the best LPIPS (0.079) at 182 ms/frame, and Multi–Pass achieved comparable perceptual quality (LPIPS 0.080) at higher computational cost. Aggressive was the least efficient, with a slightly higher LPIPS (0.084) despite five times the runtime of Dilated. Across all mask configurations, SSIM variations remained within 0.01, confirming the sufficiency of the YOLOv8s+SAM-ViT-B mask.

Averaged across videos, OpenCV Telea delivers the best quantitative scores and fastest runtimes, making it a strong baseline when speed and metric performance are priorities. However, qualitative comparisons demonstrate that LaMa (especially Multi–Pass) offers superior texture and structural coherence in challenging background regions. This highlights a discrepancy between traditional metrics and human perception: simpler interpolation methods can outperform learning-based models on SSIM and LPIPS, while still lacking the globally coherent inpainting that visually stands out in LaMa outputs. Future work should consider combining both approaches or developing perceptual metrics that better align with human judgment.

5.5. Character Insertion Evaluation

Here, we perform the actual prompt-based character insertion to replace the original human in the video. Since we operate on a frame-by-frame basis using a Stable Diffusion model, our primary objective is to minimize temporal inconsistencies across consecutive frames. Figure 4 illustrates an example output from our initial setup. While the generated character correctly mimics the human pose, the clothing and stylistic details fluctuate over time, indicating poor temporal coherence. To address this, we systematically tune the CFG guidance and strength values to reduce perceptual drift and improve consistency. Additionally, we incorporate a character-specific LoRA model to reinforce stylistic fidelity throughout the sequence. The generated character is composited onto a high-quality inpainted background produced by the YOLOv8 + SAM-ViT + LaMa pipeline.

5.5.1 Hyperparameter tuning on CFG Guidance and Strength

To determine the optimal configuration for maximizing temporal consistency, we conducted a hyperparameter search over the CFG guidance scale and the strength parameter. For each setting, we computed the LPIPS score across all video frames and reported both the mean and

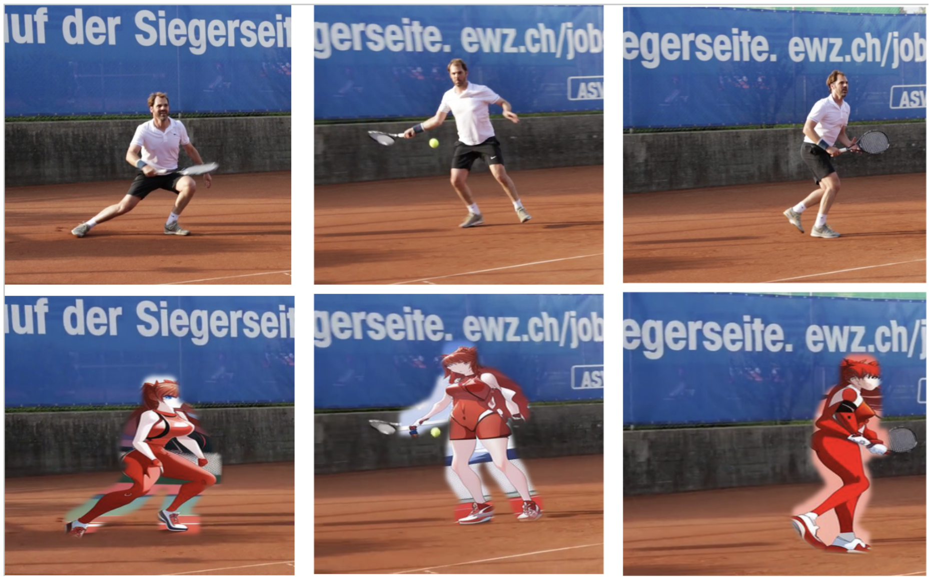
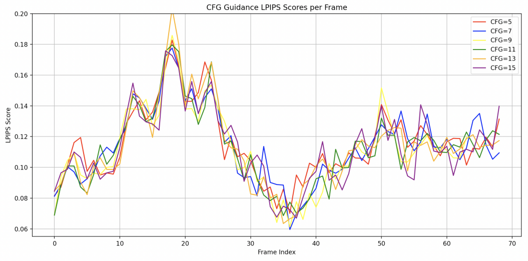


Figure 4. Here we are replacing the tennis player with the anime character Asuka Soryu Langley from the renowned anime Neon Genesis Evangelion. The drawn character appears to have the correct pose however the clothing and facial expressions appear to slightly change.

variance in the tables below. Lower LPIPS scores indicate higher perceptual similarity between adjacent frames, and thus better temporal coherence. Additionally, we plotted the LPIPS score across time for different CFG and strength values. These frame-by-frame graphs visualize how temporal consistency evolves throughout the video. Larger spikes correspond to greater perceptual drift between consecutive frames.

CFG Guidance Scale	Mean LPIPS ↓	Variance ↓
5	0.115566	0.000523
7	0.114034	**0.000567**
9	**0.111701**	0.000670
11	0.112750	0.000633
13	0.113883	0.000706
15	0.133575	0.000583

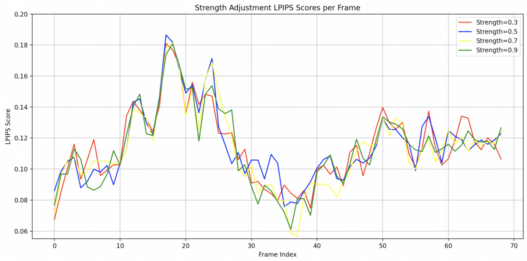
Table 6. LPIPS mean and variance for different CFG guidance values.



Here we can see the best CFG Guidance value which gives the lowest LPIPS result is equal to 9.

Strength	Mean LPIPS ↓	Variance ↓
0.3	0.114631	0.000551
0.5	0.115415	**0.000545**
0.7	**0.113021**	0.000639
0.9	0.113372	0.000616

Table 7. LPIPS mean and variance for different strength values.



Here we can see the best Strength value which gives the lowest LPIPS result is equal to 0.7.

5.5.2 Addition of LoRA model

Qualitative Evaluation of LoRA: To further improve temporal consistency, we integrate a fine-tuned LoRA model into our pipeline and evaluate its effects both qualitatively and quantitatively. As shown in Figure 5, although the LoRA-generated characters exhibit increased visual noise and some degradation in facial structure, they demonstrate more stable clothing patterns and background consistency across frames. Despite the marginal drop in per-frame visual fidelity, LoRA appears to generate outputs with enhanced temporal coherence. These artifacts could poten-

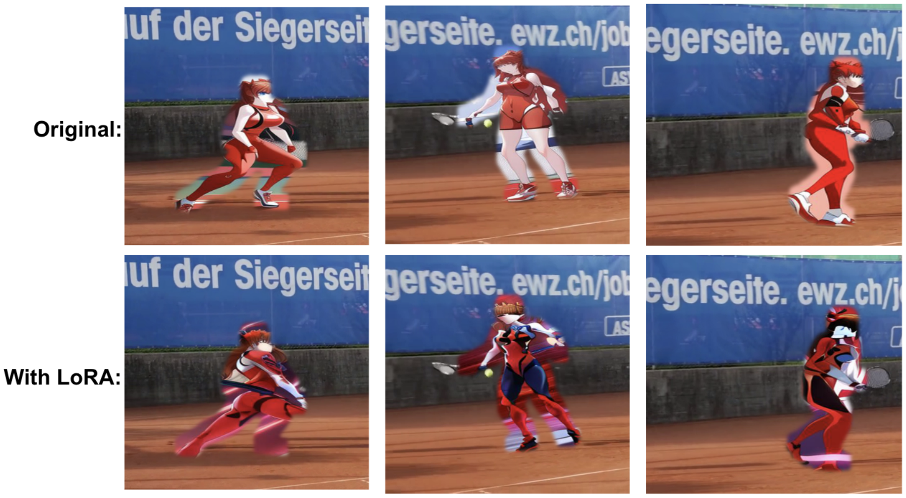
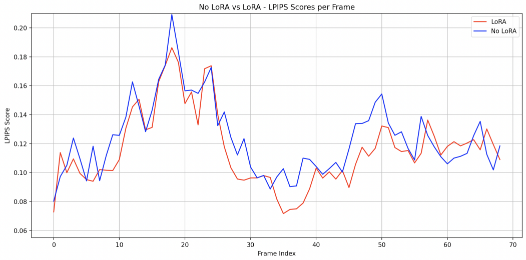


Figure 5. Here we see the character inpainted frames with and without the additon of the LoRA method.

tially be mitigated by using a higher-quality base Stable Diffusion model or switching to a more robust LoRA checkpoint.

Quantitative Evaluation of LoRA: To assess the effect of LoRA on perceptual temporal similarity, we compute LPIPS scores frame-by-frame for videos generated with and without LoRA. As with our previous hyperparameter analysis, we report the mean and variance of the LPIPS scores across the entire video sequence. The video generated without LoRA yielded a higher mean LPIPS of **0.1237** and variance of **0.000633**, while the video generated with LoRA showed a lower mean LPIPS of **0.1158** and slightly lower variance of **0.000661**. These results suggest that LoRA contributes to more perceptually consistent outputs over time. As shown in the accompanying plot, the LPIPS curve for the baseline (blue) exhibits greater variability compared to the smoother profile of the LoRA-enhanced video (red).



6. Conclusion / Future Work

In this project, we demonstrate that a carefully orchestrated zero-shot pipeline, composed entirely of off-the-shelf models, can achieve high-quality human removal and stylized character replacement in real-world videos. Our modular design enables rapid experimentation and swapping of detection, segmentation, inpainting, and generative components, allowing us to identify combinations that balance computational efficiency with perceptual realism. Through our ablation studies, we found that the combination of YOLOv8s, SAM-ViT-B, and LaMa Multi-Pass inpainting yielded the most visually consistent and efficient background restoration.

For the stylized character insertion task, we leveraged pose-guided ControlNet generation and LoRA fine-tuning to improve visual coherence. Our results show that careful tuning of CFG guidance scale and strength can significantly impact perceptual stability across frames. Incorporating a character-specific LoRA model further enhanced temporal consistency, even when overall frame quality slightly decreased.

Looking ahead, we aim to extend this work in several directions. First, we plan to incorporate multi-person tracking to support videos with multiple human subjects. Second, we intend to explore video diffusion models that better capture temporal dynamics across frames. In conjunction with this, we hope to integrate pose tracking over time as an additional conditioning signal to preserve motion consistency and character coherence. Finally, we envision building an interactive tool that allows users to upload their own videos and perform fully automated human removal and stylized replacement with customizable characters.

References

[1] stablediffusionapi/anything-v5. https://huggingface.co/stablediffusionapi/anything-v5. 5

[2] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using

part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 2

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *ECCV*, 2018. 2

[4] X. Gu, X. Wang, S. Bai, and X. Zhang. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *ICCV*, 2017. 2

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, 2021. 1

[7] G. Jocher, A. Chaurasia, J. Borovec, and et al. Ultralytics yolov8, 2023. https://docs.ultralytics.com. 1, 2, 3

[8] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 2, 4

[9] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Chang, M. G. Yong, F. Lee, S. Mennicken, A. Saluja, A. Pope, et al. Mediapipe: A framework for building perception pipelines. In *Proceedings of the 2020 ACM Multimedia Systems Conference*, pages 83–94. ACM, 2020. 2

[10] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. V. Gool. The 2017 davis challenge on video object segmentation. In *arXiv preprint arXiv:1704.00675*, 2017. 2

[11] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, volume 28, 2015. 2

[13] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, 2022. 1, 2

[14] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2089–2098, 2022. 1, 2

[15] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004. 2

[16] wsj1995. Lora models - anime character weights. `https://huggingface.co/wsj1995/LORA`, 2024. Accessed: 2025-06-04. 5

[17] Q. Yang, X. Xu, et al. Track anything: Segment and track any object in videos with foundation models, 2023. `https://github.com/gaomingqi/Track-Anything`. 2

[18] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1, 2

[19] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. `https://github.com/lllyasviel/ControlNet`, 2023. Accessed: 2025-06-04. 5