Towards General-Purpose Object Replacement in Consumer Video Editors

Omar Nasser[1] **and Youssef Badry**[2]

[1]Department of Computer Science, Minia University, Taha Hussein Street, Minya 61519, Egypt

[2]Department of Information Systems, Zagazig University, Al Ahrar Road, Zagazig 44519, Egypt

ABSTRACT Object replacement in video editing aims to remove or substitute visible entities while preserving the surrounding scene, motion, and temporal coherence. Consumer video editors increasingly expose users to tools for masking, tracking, and basic visual effects, yet object replacement remains constrained to narrow classes of objects, limited motion, or heavily supervised workflows. A general-purpose capability requires that the system handle arbitrary objects, diverse appearance, complex motion, and challenging occlusions, while remaining interactive on commodity hardware and accessible to non-expert users. This paper studies a formulation of general-purpose object replacement suitable for integration into consumer video editors and analyzes the representational, algorithmic, and system-level requirements imposed by this setting. The approach is based on layered spatiotemporal decompositions of video, learned generative models for object appearance conditioned on context, and optimization procedures that enforce temporal and geometric consistency while allowing efficient control from simple user inputs. The paper discusses how video can be factored into background, target object, and residual components; how temporal correspondence and depth ordering can be estimated robustly; and how generative editing operators can be constrained to satisfy physical and perceptual plausibility. A set of numerical considerations related to optimization, discretization, and resource constraints is described, together with implications for user interface design and failure handling. The discussion highlights trade-offs between generality, control, and efficiency, and suggests directions for extending object replacement beyond single foreground entities to richer scene-level manipulations in mainstream editing software.

1. Introduction

Digital video editing has progressively shifted from specialized professional pipelines to widely accessible consumer applications deployed on desktops, mobile devices, and web platforms [1]. Users expect tools that are simple, responsive, and robust across a wide variety of content, ranging from handheld footage to screen captures and stylized clips. Within this context, object replacement is a particularly demanding operation. It requires identifying a target object, tracking it over time, modeling its interaction with the surrounding scene, and synthesizing visually plausible replacement content that respects geometry, illumination, and motion while remaining temporally coherent. Traditional consumer tools provide masking, keying, and basic tracking, but they typically require substantial manual intervention and do not generalize to arbitrary object categories or unconstrained motions.

General-purpose object replacement seeks to approximate the flexibility of professional visual effects pipelines within the constraints of consumer software. A user should be able to identify an object by simple means such as clicking, brushing, or describing it textually, specify a replacement that might be a static image, a different object instance, or an abstract visual instruction, and obtain an edited video where the replacement behaves consistently across frames. The edited video should preserve the original camera motion, global lighting, and interactions with other objects, and it should avoid artifacts such

Article info:
Omar Nasser and Youssef Badry. *Towards General-Purpose Object Replacement in Consumer Video Editors*. Helikonpress. Publishing.
Licensed under Attribution - NonCommercial - No Derivatives 4.0 International (CC BY-NC-ND 4.0)

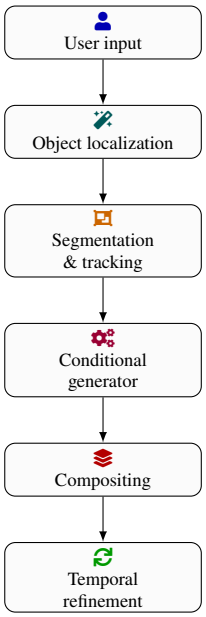


Figure 1 End-to-end object replacement pipeline: user input initializes object localization, masks are propagated over time, a conditional generator synthesizes replacement content, and compositing with temporal refinement yields the edited video sequence.

as temporal flicker, broken shadows, or inconsistent occlusions. Achieving this goal requires a synthesis of advances in scene understanding, generative modeling, motion estimation, and interactive systems [2].

The challenges are compounded by the computational and usability constraints typical of consumer environments. Offline film production workflows can rely on multi-stage manual annotation, extensive computational resources, and long processing times. By contrast, consumer editors must provide feedback in seconds, not hours, and must operate on devices with limited memory and energy budgets. This setting limits the complexity of the models that can be deployed, encourages approximate but robust algorithms, and requires careful design of user interactions so that a small amount of input can guide the system effectively. The same system must also cope with diverse content, including videos with motion blur, sensor noise, compression artifacts, or stylized color grading [3].

This paper considers the problem of general-purpose object replacement from a modeling and systems perspective. It formulates object replacement as a constrained generative editing problem on layered spatiotemporal representations of video. The formulation emphasizes a decomposition of the scene into background, object, and residual layers, coupled with motion fields and depth priors that govern occlusion relationships. On top of this representation, a generative operator modifies the object layer conditioned on the original video and a user-provided specification, while regularization terms enforce temporal smoothness and coherence with physical constraints such as approximate lighting and geometry. The approach is designed to admit approximate inference procedures suitable for consumer hardware, for example through patch-wise processing and low-rank approximations of feature tensors.

The paper also discusses learning and numerical aspects that arise when training such systems on large corpora of videos, including probabilistic formulations of uncertainty, regularization for temporal stationarity, and the impact of discretization choices on visual artifacts. Finally, it analyzes the integration of the underlying algorithms into the end-to-end pipeline of a consumer video editor, accounting for latency budgets, interaction patterns, and mechanisms for failure recovery and user control. The resulting view is descriptive rather than prescriptive and aims to clarify design trade-offs and opportunities for general-purpose object replacement in widely used editing tools.

2. Problem Formulation and Design Space

Consider a video represented as a sequence of frames $x_1, x_2, \dots, x_T$, where each frame x_t can be treated as an element of $\mathbb{R}^{H \times W \times C}$, with height H, width W, and color channels C. It is convenient to regard the video as a tensor [4]

$$V \in \mathbb{R}^{T \times H \times W \times C},$$

with indices t, i, j, c for time, spatial position, and channel. Object replacement focuses on a specific subset of pixels associated with an object that appears across time. A binary mask sequence [5]

$$M = M_1, \dots, M_T$$

is used, with each $M_t \in \{0,1\}^{H \times W}$ indicating membership of each pixel in the target object. In practice, M is unknown and must be estimated from minimal user input and the video itself.

The user provides a replacement specification s that encodes the desired appearance and behavior of the substituted object. The specification may consist of an image, a short video clip, a text description, or a parametric representation such as a three-dimensional mesh with materials. From a mathematical standpoint, [6] s is an element of an abstract space $\mathcal{S}$ of control

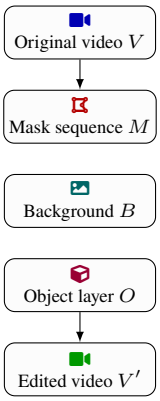


Figure 2 Layered compositing view of a clip: the original video is decomposed into a time-varying mask, background layer, and object layer. Replacement modifies only the object layer while preserving background content, and the layers are recombined to produce the edited sequence.

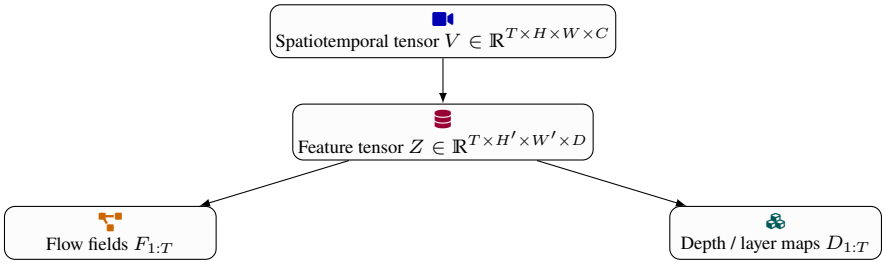


Figure 3 Spatiotemporal scene representation: a video tensor is mapped into a lower-resolution feature tensor, from which optical flow fields and approximate depth or layer maps are derived to capture motion and occlusion structure over time.

variables, which may be high dimensional and heterogeneous. The goal of object replacement is to produce an edited video V' such that the following conditions approximately hold. First, outside the object support defined by M, the edited video should match the original:

$$V'_{t,i,j,c} = V_{t,i,j,c} \quad \text{for} \quad M_t i, j = 0.$$

Second, within the object region, the content of V' should correspond to a plausible realization of the replacement specified by s, conditioned on contextual information from the original video. Third, the edited video should be temporally coherent, avoiding abrupt changes in appearance or motion.

A convenient way to express these requirements is via a layered decomposition. The video is modeled as a composition of a background layer B and an object layer O, with the object mask [7] M acting as an alpha matte. For each time t and pixel i, j,

$$x_t i, j = M_t i, j \, o_t i, j \;\; 1 - M_t i, j \, b_t i, j,$$

where $b_t i, j$ and $o_t i, j$ denote background and object colors. The edited video x'_t retains b_t but replaces o_t with a synthesized object color $\tilde{o}_t i, j$ derived from s and the surrounding context. This leads to

$$x'_t i, j = M_t i, j \, \tilde{o}_t i, j \;\; 1 - M_t i, j \, b_t i, j.$$

In a linear algebraic view, for each time [8] t, one can flatten the frame into a vector $x_t \in \mathbb{R}^N$ with $N = HWC$, and interpret the mask as a diagonal matrix $D_t \in \mathbb{R}^{N\times N}$ with entries 0 or 1. The compositing equation becomes

$$x_t = D_t \, o_t \;\; I - D_t \, b_t,$$

where I is the identity matrix. This representation emphasizes that object replacement can be seen as solving for $\tilde{o}_t$ given x_t, an estimate of D_t, and a constraint that b_t should remain consistent with the original video.

In a probabilistic formulation, the edited video is modeled as a random variable V' conditioned on the original video V and the specification s. The system defines a conditional distribution [9]

$$p_\theta V' \mid V, s,$$

with parameters θ capturing learned priors over appearance, motion, and object behavior. Deterministic editing operators correspond to selecting a single point estimate, for instance

$$V' = \arg\max_U \; p_\theta U \mid V, s,$$

or $V' = \mathbb{E}V' \mid V, s$, depending on the loss and optimization strategy. In practice, consumer tools often expose a small number of stochastic samples or a deterministic edit controlled by user-adjustable strength parameters.

The design space of general-purpose object replacement includes several axes. The first concerns the degree of structural modeling of the scene, ranging from purely two-dimensional approaches that operate in image space to more explicit three-dimensional representations with estimated depth or dynamic

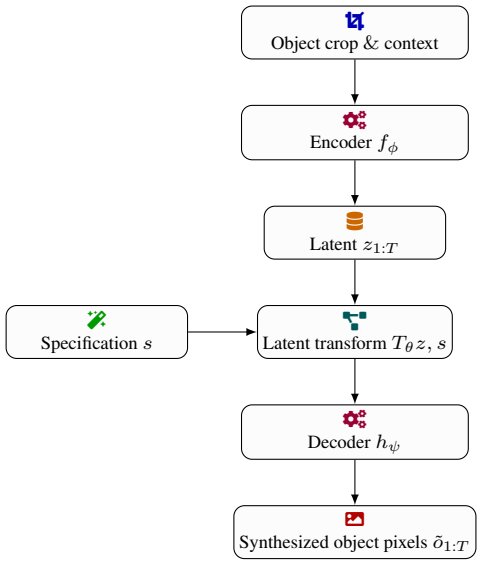


Figure 4 Latent-space conditional generator: encoder features for the object and its context are mapped to latent variables, transformed according to a user specification, and decoded into synthesized object pixels that can be composited back into the video.

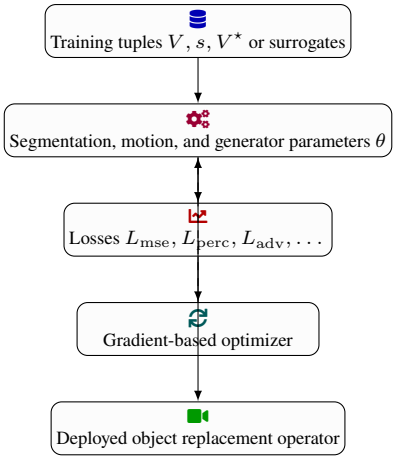


Figure 5 Learning and numerical loop: datasets of clips and specifications drive parameter updates for segmentation, motion estimation, and generative components via composite losses and gradient-based optimization, resulting in a deployed operator used at inference time.

geometry. The second axis concerns the nature of the generative operator applied to the object region, which might be patch-based, latent-space based, or defined through iterative stochastic dynamics. A third axis involves the level of user control and interpretability, including how the specification s is represented and how constraints such as preserving object pose or motion are enforced. Finally, the system must choose an appropriate trade-off between accuracy and computational cost, determining how much of the video can be processed at full resolution and how aggressively to downsample or approximate components of the optimization.

Within this design space, the formulation adopted here emphasizes layered models with explicit masks, background preservation, and conditional generative models that act primarily on the object layer while receiving rich context from the background and temporal neighbors [10]. This decomposition reflects the goal of integrating the method into consumer editors, where preserving original content outside the target object is generally desirable, and where users expect spatially localized control without needing to reason about the entire scene.

3. Spatiotemporal Scene and Object Representation

A central choice in object replacement is the representation of the video and the target object. A naive representation that treats each frame independently cannot capture the temporal structure and motion constraints that govern realistic edits. Instead, it is useful to view the video as a spatiotemporal tensor and to impose structure through factorization and auxiliary fields.

At the lowest level, the video is modeled as a tensor $V \in \mathbb{R}^{T\times H\times W\times C}$. A feature encoder defines a mapping

$$f_\phi : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H'\times W'\times D},$$

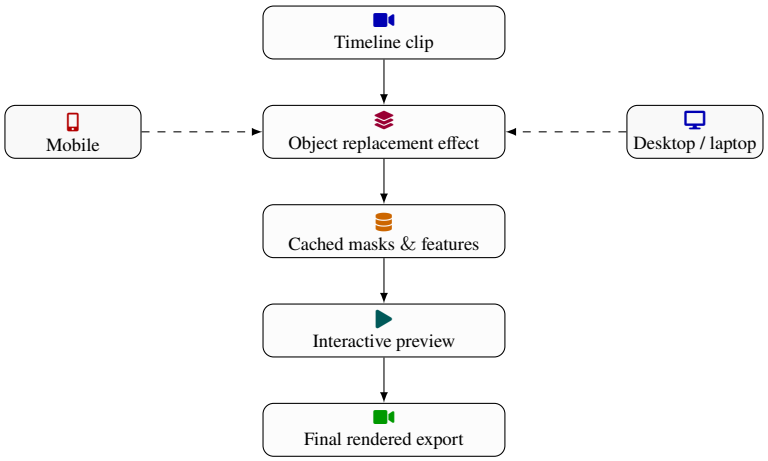


Figure 6 Integration into a consumer editor: object replacement is attached as an effect on a timeline clip, maintains intermediate caches for interactive preview, and adapts its implementation to heterogeneous devices such as mobile and desktop systems before producing the final render.

Symbol	Meaning	Space / Dimensionality
V	Full video sequence	$\mathbb{R}^{T\times H\times W\times C}$
x_t	Frame at time t	$\mathbb{R}^{H\times W\times C}$
M_t	Binary object mask at frame t	$\{0,1\}^{H\times W}$
s	User replacement specification (text, image, mesh, . . .)	Abstract control space $\mathcal{S}$
B, O	Background and object layers in compositing	Same shape as x_t
Z	Spatiotemporal feature tensor	$\mathbb{R}^{T\times H'\times W'\times D}$

Table 1 Core notation used in the formulation of general-purpose object replacement.

with parameters ϕ, that converts each frame into a feature grid. Applying this encoder across time yields a tensor

$$Z \in \mathbb{R}^{T\times H'\times W'\times D},$$

which captures semantic and geometric information in a compact form. The tensor Z can be decomposed into components associated with the background and the object. For instance, one can write

$$Z = Z^{\mathrm{bg}} \; Z^{\mathrm{obj}},$$

with Z^{bg} capturing static or slowly varying parts of the scene and [11] Z^{obj} concentrated around the object mask support. This decomposition can be enforced through attention mechanisms or through explicit multiplication by upsampled masks.

A useful abstraction is to represent object motion through a sequence of flow fields

$$F = F_1, \ldots, F_T,$$

where each $F_t \in \mathbb{R}^{H\times W\times 2}$ encodes a displacement from frame t to frame t 1 in pixel coordinates. For positions inside the object region, F_t describes the motion of object pixels, while for the background it approximates camera or scene motion. The composition of flow fields over time defines trajectories through the image domain. For a pixel index i, j in frame [12] t, one can define a trajectory

$$\gamma_{t,i,j}k = i_k, j_k,$$

for $k \geq t$, obtained by integrating the discrete flow forward. A continuity constraint encourages trajectories inside the object mask to remain within the object region as time evolves.

Conceptually, object replacement benefits from some notion of depth or layer ordering. A coarse depth map

$$D_t \in \mathbb{R}^{H\times W}$$

associates each pixel with an approximate distance from the camera. Even when D_t is not metrically precise, it can help resolve which surfaces should occlude others when the object is replaced by new geometry. One can model the scene as a set of layers indexed by depth intervals and assign the object to one or more such layers. A discrete representation arises by assigning each pixel a layer index

$$\ell_t i, j \in \{1, \ldots, L\},$$

Design axis	Representative options	Typical trade-offs
Scene structure	Purely 2D image-space vs. layered 2D vs. explicit 3D	Geometric consistency vs. compute and memory cost
Generative operator	Patch-based, latent diffusion, iterative refinement	Local detail vs. global coherence and controllability
User control	Clicks/brushes, prompts, reference media	Simplicity of UI vs. expressiveness of specification s
Motion treatment	Per-frame edits vs. flow- or trajectory-aware edits	Runtime vs. temporal coherence and flicker suppression
Accuracy vs. cost	Full-resolution, full-sequence optimization vs. approximations	Visual fidelity vs. latency and device constraints

Table 2 High-level design axes for object replacement systems and their qualitative trade-offs.

Representation	Auxiliary fields	Suitability for consumer editors
Per-frame 2D	None or simple features	Lowest cost, limited temporal/geometry consistency
Layered 2D + masks	Binary or soft masks M_t	Good localization, straightforward compositing
2D + flow/depth	Optical flow F_t, depth D_t or layer indices ℓ_t	Better occlusions and temporal coherence at moderate cost
Dynamic radiance field	Continuous $F_\psi x, t$ in 3D	Strong geometric guarantees, often too heavy for typical devices
Hybrid features	Low-res $Z^{\mathrm{bg}}, Z^{\mathrm{obj}}$ plus flow/depth	Balances semantics, motion, and efficiency for interactive use

Table 3 Spatiotemporal scene and object representations considered for consumer-oriented deployment.

for some small number L of layers. Occlusion relationships are then determined by ordering these layers along the viewing direction [13].

For richer modeling, it is natural to consider representations that approximate a dynamic three-dimensional scene. Let $X \subset \mathbb{R}^3$ denote the spatial domain and $1, T$ the discrete time interval. A continuous representation associates each point $x, t \in X \times 1, T$ with color and density through a function

$$F_\psi : X \times 1, T \to \mathbb{R}^4,$$

parameterized by ψ, where the output encodes radiance and opacity. Rendering a frame involves integrating along camera rays. For a camera with projection Π_t, a pixel [14] i, j corresponds to a ray

$$r_{t,i,j} u = o_t \;\; u \, d_{t,i,j},$$

with origin o_t and direction $d_{t,i,j}$. The rendered color is given by an integral of the form

$$x_t i, j = \int_0^\infty wu \, cr_{t,i,j} u, t \, \mathrm{d}u,$$

where c denotes color along the ray and wu weights contributions according to density and transmittance. In discrete implementations, this integral is approximated by a finite sum over samples along each ray. Object replacement then amounts to modifying F_ψ within a spatial region associated with the object while preserving consistency with surrounding rays.

While fully three-dimensional formulations provide strong guarantees on geometric consistency, they are computationally demanding and may not be suitable for typical consumer hardware. A compromise is to maintain a primarily two-dimensional representation augmented with auxiliary fields such as depth, flow, and feature tensors that encode partial three-dimensional information. Within this setting, the object is represented by its mask sequence M, its feature embedding extracted from [15] Z, and optionally its approximate depth distribution. Replacement operations operate on these representations, synthesizing new features $\tilde{Z}^{\mathrm{obj}}$ conditioned on the specification s and the background features Z^{bg}. Decoders then map these features back to pixel space to obtain V'.

Regularization on the spatiotemporal representation is important for ensuring stable and coherent edits. A standard approach is to define an energy functional over the feature tensor Z and mask sequence M. For example, one can consider a term

$$E_{\mathrm{smooth}} = \lambda_t \sum_{t=1}^{T-1} \|Z_{t1} - Z_t\|_2^2 \;\; \lambda_s \sum_{t=1}^{T} \sum_{p,q \in \mathcal{N}} \|Z_{t,p} - Z_{t,q}\|_2^2,$$

Pipeline stage	Main inputs	Primary outputs / responsibilities
Initialization	User interactions u, reference frame x_{t_0}	Seed mask M_{t_0}, initial object likelihoods
Segmentation & tracking	Features Z, motion cues, seed mask	Mask sequence $M_{1:T}$, confidence maps C_t
Motion/depth estimation	Video V, features Z	Flow fields F_t, depth maps D_t or layer indices
Replacement synthesis	Context $x_{1:T}^{\text{ctx}}$, masks, spec s	Synthesized object pixels $\tilde{o}_{1:T}$ or features $\tilde{Z}^{\text{obj}}$
Compositing & refinement	Original V, masks, $\tilde{o}_{1:T}$, shading S_t	Edited video V' with temporal smoothing and artifact suppression

Table 4 End-to-end object replacement pipeline from user input through synthesis and compositing.

Strategy	Latent / state structure	Temporal coherence mechanism
Deterministic operator	Direct mapping $V, s \mapsto V'$	Implicit sharing of parameters across frames
Conditional latent model	Latents z_t from encoder g_ϕ	Markov priors $pz'_t \mid z'_{t-1}$, smoothness penalties
Iterative refinement	States $\tilde{o}_t^k$ updated by Δ_θ	Shared refinement steps over time and iterations
Stochastic sampling	Noise-driven chain $z_t'^k$	Coupled transitions and truncated chains for speed
Patch/trajectory-based	Local patches or flow-aligned tubes	Aggregation along motion trajectories $\gamma_{t,i,j}$

Table 5 Representative generative editing strategies for synthesizing replacement object content.

where $\mathcal{N}$ denotes a set of neighboring pixel pairs. This energy encourages both temporal and spatial smoothness in feature space, which can reduce flicker and protect against spurious high-frequency artifacts introduced by generative models. In matrix form, such smoothness can be related to discrete Laplacian operators on spatiotemporal graphs.

The combination of tensors, flow fields, and mask sequences provides a flexible yet compact representation for general-purpose object replacement [16]. It supports a variety of generative operators while allowing motion and depth information to propagate constraints across time. The representation also aligns with the constraints of consumer editors, as feature tensors can be maintained at lower resolution than the original video while still capturing high-level semantics and geometry.

4. Object Replacement Pipeline and Algorithms

Given a representation for the video and object, an object replacement system must connect user input, scene analysis, generative editing, and compositing into a coherent pipeline. In a consumer editor, this pipeline must operate under latency constraints and with minimal manual intervention. Conceptually, it can be decomposed into stages: object localization and initialization, segmentation and tracking, replacement synthesis, and compositing with temporal refinement. Each stage must handle uncertainty and errors gracefully, as misestimation in early stages can propagate.

Object localization and initialization begin with user input. The user might click on a point inside the target object in a reference frame, provide a rough brush stroke, or offer a textual description. This input is converted into an initial mask estimate M_{t_0} for some frame t_0. A feature encoder maps the frame to a feature grid Z_{t_0}, and a classifier or matching function distinguishes object features from background. One can model the probability that a pixel p belongs to the object as [17]

$$PM_{t_0}p = 1 \mid Z_{t_0}p, u,$$

where u encodes the user input. A logistic regression or small neural head operating on $Z_{t_0}p$ and u estimates these probabilities. At this stage, the system might present the user with a preview mask, allowing quick corrections that refine M_{t_0} and implicitly adjust model parameters for the current clip.

Segmentation and tracking extend the mask M_{t_0} to the entire sequence. One approach constructs a spatiotemporal graph where nodes correspond to pixels or patches across frames, and edges connect nodes that are likely to correspond based on

Learning setting	Data assumption	Typical objective components
Fully supervised	Pairs $V, s, V^\star$ with ground truth edits	Pixel losses L_{mse}, perceptual and adversarial terms
Synthetic supervision	Edits generated by procedural transforms	Consistency under known transformations, reconstruction fidelity
Self-supervised	Unlabeled videos only	Masked prediction, inpainting, motion-consistency tasks
Unsupervised generative	Latent-variable modeling of clips	Variational lower bounds, likelihood maximization
Hybrid / fine-tuning	Mix of the above on diverse corpora	Multi-task objectives with task-specific weights

Table 6 Learning regimes for training segmentation, motion, and generative components of the system.

Aspect	Numerical / resource issue	Mitigation in the proposed setting
Precision	Rounding error and cancellation in low-bit formats	Normalization, clipping, scaled losses, mixed-precision policies
Temporal discretization	Over/under-smoothing due to fixed Δt	Motion-aware regularization and adaptive temporal weights
Attention and features	Quadratic cost in $H'W'$ and T	Local attention, low-rank approximations, feature downsampling
Truncated sampling	Bias from short Markov chains	Empirical step budgets, biased kernels, deterministic shortcuts
Memory limits	Storing tensors for long, high-res clips	Tiling with overlaps δ, caching, streaming and reuse of intermediates

Table 7 Key numerical and computational considerations for interactive deployment on commodity hardware.

appearance and motion cues. Let

$$G = V_G, E_G$$

denote this graph, with vertex set V_G and edge set E_G. Edges carry weights w_{pq} reflecting similarity between nodes p and q. A label propagation scheme seeks label assignments [18]

$$y_p \in \{0, 1\}$$

for each node that minimize an energy of the form

$$E_{\text{seg}} = \sum_p \ell y_p; \hat{y}_p \; \mu \sum_{p,q \in E_G} w_{pq} \, |y_p - y_q|,$$

where $\hat{y}_p$ denotes an initial label (for example from M_{t_0} or a classifier) and ℓ is a penalization function that promotes agreement with initial labels. The second term encourages spatial and temporal smoothness of the labels along edges. Minimization of E_{seg} can be performed approximately using graph cuts or continuous relaxations, but for consumer applications it is common to employ a parametric model that approximates this energy through convolutional or attention-based operations for efficiency.

Once mask sequences M_t are established, object replacement requires synthesizing new content for the object region in each frame. Let x_t^{ctx} denote the context around the object, obtained by masking out the object region and potentially padding with neighboring pixels. A generative model defines a conditional distribution

$$p_\theta\big(o_{1:T} \mid x_{1:T}^{\text{ctx}}, s\big),$$

where o_t denote the object pixels in frame [19] t. A common choice is to operate in a latent space. An encoder maps the object and context to latent variables

$$z_t = g_\phi x_t, M_t,$$

and a decoder reconstructs object pixels from modified latents

$$\tilde{o}_t = h_\psi z_t', M_t,$$

where z_t' is a transformed latent that reflects the replacement defined by s. The transformation might be expressed as

$$z_t' = T_\theta z_t, c_t,$$

Component	Option A	Option B	Integration note
Processing schedule	Offline clip analysis	Just-in-time per visible frame	Hybrid strategies balance latency and quality
Feature resolution	Full-resolution features	Downsampled H', W' with decoding	Downsampling trades detail for responsiveness
Tiling	No tiling	Overlapping tiles with blending	Overlap δ avoids seams at tile boundaries
Effect ordering	Replace then grade	Grade then replace	Choice affects stability and creative control
Model variant	Compact mobile model	Larger desktop model	Resource-aware selection via θ^k families

Table 8 System-level integration choices for deployment in multi-effect consumer video editors.

Challenge	Consequence for edits	Example mitigation mechanisms
Ambiguous object identity	Inconsistent masks and tracking	Interactive corrections, uncertainty maps C_t, robust energies
Underspecified intent	Plausible but unwanted object variants	Multiple samples, interpretable controls on pose, style, and strength
Data and model bias	Systematic failures on certain content	Diverse training corpora, monitoring, domain adaptation
Potential misuse	Deceptive or harmful manipulations	Provenance metadata, watermarking, UI indicators for synthetic content
Limited interpretability	Hard-to-predict model behavior	Visualizing masks, flows, depth, and intermediate layers to users

Table 9 Selected limitations and associated mitigation strategies relevant to responsible object replacement.

with c_t encoding specification features derived from s and the context x_t^{ctx}. In a stochastic setting, one can model

$$z'_t \sim q_\theta z \mid z_t, c_t,$$

and sample multiple realizations to provide the user with different replacement options.

Temporal coherence is encouraged by coupling latent variables across time. One can introduce a prior of the form

$$pz'_{1:T} = \prod_{t=1}^{T} pz'_t \mid z'_{t-1},$$

with a Markovian dependency that penalizes large variations in consecutive latents. For example, [20]

$$pz'_t \mid z'_{t-1} \propto \exp\bigl(-\alpha \, \|z'_t - Az'_{t-1}\|_2^2\bigr),$$

where A is a linear operator that propagates latent dynamics and α controls temporal smoothness. In high-resolution videos, one often approximates this propagation through local convolutions or attention with restricted temporal windows to maintain computational feasibility.

An alternative path uses iterative refinement. Starting from an initial guess $\tilde{o}_t^0$ obtained by simple inpainting or copy-and-paste, an update operator

$$\tilde{o}_t^{k1} = \tilde{o}_t^k \; \Delta_\theta\bigl(\tilde{o}_t^k, x_t^{\mathrm{ctx}}, M_t, s\bigr)$$

progressively improves object appearance [21]. The update can be driven by a learned gradient that approximates the negative gradient of a perceptual or adversarial loss, combined with temporal regularization terms. When implemented with shared weights over time, this procedure encourages consistent refinement patterns across frames and can be truncated after a small number of iterations to satisfy runtime constraints.

Compositing combines the synthesized object pixels $\tilde{o}_t$ with the original background. In the simplest case, the compositing equation

$$x'_t = M_t \odot \tilde{o}_t \; 1 - M_t \odot x_t$$

is sufficient, where $\odot$ indicates elementwise multiplication with broadcasting over color channels. However, realistic replacement often requires adjustment of lighting, shadows, and reflections. These effects can be approximated through additional layers or by predicting adjustments to the background in regions neighboring the object. For example, a shading adjustment S_t can modulate

the background as

$$x'_t = M_t \odot \tilde{o}_t \; 1 - M_t \odot S_t \odot x_t,$$

with S_t close to one outside a narrow band around the object boundary.

Finally, post-processing steps aim to enforce consistency and reduce artifacts [22]. A temporal smoothing filter can be applied to the edited frames. Let $\hat{x}'_t$ denote the initially composited frames. A smoothed sequence

$$x''_t = \frac{\sum_{k=-K}^{K} w_k \hat{x}'_{tk}}{\sum_{k=-K}^{K} w_k},$$

with weights w_k summed over frames within a window of size $2K$ 1, reduces high-frequency temporal variation. The weights can be adapted per pixel based on estimated motion or confidence in the generative predictions, leading to a content-aware temporal filter. This smoothing must be kept lightweight to meet performance constraints.

The described pipeline connects discrete and continuous formulations of segmentation, generative modeling, and compositing. Its components can be implemented with different architectural choices and training regimes, but they share the objective of mapping from the original video and user specification to an edited video that is visually stable and respects the original scene structure.

5. Learning, Inference, and Numerical Aspects

Training and deploying a general-purpose object replacement system involve probabilistic modeling, multivariate optimization, and numerical approximations that impact visual quality and performance. The core objective is to learn parameters θ for segmentation, motion estimation, and generative synthesis components so that the resulting conditional distribution [23] $p_\theta V' \mid V, s$ aligns with user expectations across diverse content.

In a supervised setting, training data may consist of tuples $V, s, V^\star$, where $V^\star$ is a ground truth edited video that reflects the application of specification s to original video V. Often, such pairs are unavailable at scale, motivating proxy objectives constructed from synthetic transformations or self-supervised tasks. Regardless of data source, a common principle is to minimize an expected loss of the form

$$\mathcal{L}\theta = \mathbb{E}\big[L V', V^\star\big],$$

where $V' \sim p_\theta \cdot \mid V, s$. The loss L can include pixelwise terms, feature-space discrepancies, and adversarial or perceptual metrics. A pixelwise mean squared error can be expressed as

$$L_{\text{mse}} = \frac{1}{THWC} \sum_{t\,i,j,c} \big(x'_t i, j, c - x^\star_t i, j, c\big)^2.$$

This term penalizes deviations from ground truth and encourages basic photometric fidelity [24]. However, it often underestimates perceptual quality, motivating feature-based distances computed in the embedding space of a fixed network.

When explicit ground truth is unavailable, one can adopt a self-supervised strategy. For instance, one can sample an object region in a frame, apply a synthetic transformation such as color change or texture replacement, and train the system to reconstruct the transformed region given the original context and specification encoding this transformation. This defines a surrogate distribution where $V^\star$ is algorithmically generated. Alternatively, one can train generative components with unsupervised objectives such as maximizing the log-likelihood of observed video patches under a latent variable model. If z denotes latent variables, the marginal likelihood

$$\log p_\theta x = \log \; p_\theta x \mid z \, p_\theta z \, \mathrm{d}z$$

is usually intractable, but can be lower bounded by variational approximations that introduce an inference model $q_\phi z \mid x$. Optimization proceeds by maximizing this lower bound via gradient-based methods.

Inference during editing must respect tight time and memory budgets. For segmentation, one can precompute features Z at a reduced resolution and propagate masks using lightweight recurrent or convolutional modules. The numerical complexity of these operations is typically $\mathcal{O}THW'D^2$ for naive dense attention patterns, where [25] H', W', and D denote feature dimensions. To avoid quadratic dependencies on spatial resolution, attention can be localized or replaced with convolutional filters. The choice of numerical scheme influences not only speed but also stability, as under-parameterized models may fail on challenging motion patterns.

For generative synthesis, stochastic sampling procedures must be carefully truncated. Suppose synthetic object latents z'_t are obtained by iterating a Markov chain. Let

$$z'^{k1}_t = G_\theta\big(z'^k_t, \eta^k, c_t\big),$$

where η^k are random noise variables. In theory, sampling until convergence yields accurate approximations to the target distribution. In practice, only a small number of steps can be performed. Numerical analysis of such truncated chains involves examining the spectral properties of the transition operator and bounding the mixing time. In consumer settings, one often chooses a fixed number of iterations that empirically balances quality and speed, without guaranteeing full convergence [26]. To mitigate the resulting bias, one can bias the transition kernel to reduce variance or incorporate deterministic shortcuts that approximate later iterations.

The training process must address stability and convergence. Let θ_k denote model parameters at iteration k. A gradient-based update with learning rate α_k takes the form

$$\theta_{k1} = \theta_k - \alpha_k \, \nabla_\theta \mathcal{L}\theta_k,$$

where the gradient is estimated over minibatches. Convergence properties depend on smoothness and convexity assumptions on $\mathcal{L}$, which rarely hold in high-dimensional non-linear models. Empirically, adaptive learning rate schemes and normalization layers are used to control gradient magnitudes, but these introduce additional hyperparameters whose influence on the final user experience must be monitored. From a multivariate calculus perspective, the Hessian $H\theta = \nabla^2_\theta \mathcal{L}\theta$ captures curvature and

can indicate sensitive directions where small parameter changes produce large changes in loss. Although explicit computation of H is infeasible for large models, approximate second-order information can guide learning rate schedules [27].

Uncertainty quantification plays a role in deciding when to present results to the user or request additional input. In a Bayesian interpretation, one might place a prior $p\theta$ over parameters and consider the posterior

$$p\theta \mid \mathcal{D} \propto p\mathcal{D} \mid \theta \, p\theta,$$

given data $\mathcal{D}$. Predictive uncertainty for a new video V can then be expressed as

$$pV' \mid V, s, \mathcal{D} = \; p_\theta V' \mid V, s \, p\theta \mid \mathcal{D} \, \mathrm{d}\theta.$$

Exact evaluation is intractable, but ensemble methods or approximate Bayesian techniques can provide empirical measures of variability across different parameter realizations. At inference time, the system can use these measures to detect low-confidence edits, flagging them to the user or automatically reducing the strength of generative modifications to avoid obvious artifacts.

Numerical precision is another practical concern. Consumer devices often rely on reduced-precision arithmetic such as floating point formats with 16 bits or less [28]. These formats accelerate computation and reduce memory usage but may introduce rounding errors. Consider a loss gradient g computed as the difference of large terms a and b with similar magnitude. In reduced precision, the difference $a - b$ may suffer from catastrophic cancellation, leading to high relative error. Careful normalization and rescaling of intermediate quantities can alleviate these issues by keeping values within numerically stable ranges. For example, feature vectors can be normalized to unit variance, and activations can be clipped or transformed by smooth nonlinearities that limit extreme values.

Temporal discretization also influences artifact patterns. The video is naturally discrete in frames, but motion and generative dynamics are conceptually continuous. When modeling temporal smoothness, it can be useful to approximate time derivatives with finite differences. For a quantity u_t, one can approximate the derivative as

$$\frac{\partial u}{\partial t}\Big|_t \approx 29 \frac{u_{t1} - u_t}{\Delta t},$$

where Δt is the time between frames. Regularizing this derivative to be small encourages slow temporal change. However, if Δt is large or variable, such a regularizer can oversmooth fast motion. In consumer content, frame rates and motion patterns vary, so temporal regularization must adapt to local motion estimates to avoid blurring dynamic regions while still suppressing flicker in nearly static areas.

These learning and numerical considerations interact closely with system design. Choices made for stability, convergence, and precision directly impact latency, memory usage, and visual quality. A robust object replacement system for consumer editors must thus balance model capacity, data-driven learning, and numerical approximations in a way that aligns with the constraints of typical devices and user expectations of responsiveness.

6. System Integration in Consumer Video Editors

Beyond modeling and numerical algorithms, object replacement must be integrated into the architecture and interaction patterns of consumer video editors. Such editors support multiple tracks, real-time preview, and a range of effects that can be composed in arbitrary orders. The object replacement operator is one effect among many, and it must interoperate gracefully with color grading, transitions, and other transformations [30].

A typical integration strategy treats object replacement as a clip-level effect that can be attached to a video segment on the timeline. When the user selects the effect, the editor displays controls for specifying the target object and the replacement. Internally, the system must decide when to process the clip. Offline processing on effect attachment can precompute masks, motion fields, and intermediate representations, enabling fast scrubbing and playback of the edited result. However, offline processing may introduce initial latency. Alternatively, the system can perform just-in-time processing, incrementally computing results for frames as they become visible during playback. This reduces initial delay but may lead to temporary low-quality previews while more accurate estimates are refined in the background. Given the requirement to avoid background work in some deployment contexts, it is common to adopt a hybrid approach, performing a limited amount of precomputation and then refining results when the user pauses or focuses on a specific frame.

Resource management is critical. For a clip with T frames and resolution $H \times W$, storing full-resolution intermediate tensors for all frames may exceed memory limits. Downsampling and tiling strategies address this [31]. Features can be computed at a lower spatial resolution

$$H' = \frac{H}{r}, \quad W' = \frac{W}{r},$$

with a reduction factor r chosen so that the resulting tensors fit in memory. For full-resolution editing, decoding from low-resolution features must preserve important details. Tiling partitions frames into overlapping patches processed independently. An overlap width

$$\delta$$

ensures that boundaries between tiles do not introduce seams. After processing, overlapping regions can be blended using smooth weighting functions to avoid visible discontinuities.

The user interface must expose a manageable set of controls that map to the underlying high-dimensional specification s. In many scenarios, s is derived from a simple input such as a text prompt or reference image. Internally, a text encoder or image encoder maps this input into a latent code [32]

$$c = es,$$

which conditions the generative model. The editor may allow users to adjust parameters such as the strength of replacement, degree of adherence to the original object pose, or level of

stylization. These parameters can be modeled as scalar weights $\lambda_1, \lambda_2, \ldots$ that scale different loss terms or modify conditioning signals. For instance, a control for preserving motion can correspond to a weight λ_{motion} that scales a temporal consistency term in the generative objective.

Previewing and interaction latency impose additional constraints. The editor must display updated frames at interactive rates when the user scrubs through the timeline or adjusts effect parameters. To support this, the system can cache edited frames and their dependencies. Suppose the user modifies a control that affects only the appearance of the object but not its segmentation or motion. In this case, recomputation can be restricted to the generative and compositing stages, reusing previously computed masks and flow fields. Conversely, if the user changes the object selection, segmentation and tracking must be recomputed, while generative and compositing stages can be deferred until the user stops adjusting the selection.

Robust handling of failure cases is important for user satisfaction. Segmentation errors may cause parts of the object to be omitted or background regions to be edited inadvertently [33]. Motion estimation failures can introduce ghosting or streaking. Generative models might hallucinate content that conflicts with user intent. To mitigate these issues, the editor can provide visualization tools such as overlaying the mask on the video or displaying uncertainty maps that highlight regions of low confidence. Internally, binary masks can be accompanied by soft confidence maps

$$C_t i, j \in 0, 1,$$

representing the system's belief that each pixel belongs to the object. During compositing, these confidences can soften transitions between edited and original content, reducing abrupt artifacts in uncertain regions.

The editor must also integrate object replacement with other effects in a compositional pipeline. Effects may be ordered so that object replacement happens before or after color adjustments, geometric transforms, or overlays. Because generative replacement is often sensitive to color and contrast, applying it in the raw color space before stylization can improve stability. On the other hand, certain creative workflows may prefer to stylize first and then replace objects within the stylized appearance. To support both, the effect graph can allow the object replacement node to be placed at different stages, with internal conversions handling changes in color space or resolution. From an implementation perspective, effect nodes define operators

$$\mathcal{E}_k : \mathcal{X} \to \mathcal{X},$$

on the space [34] $\mathcal{X}$ of clips, and the overall pipeline composes these operators. Object replacement corresponds to a particular $\mathcal{E}_{\text{obj}}$, which must be associative with respect to other operators in a way that preserves intuitions about ordering.

Finally, system integration must consider portability. Consumer editors run on heterogeneous hardware, including smartphones, tablets, and desktops with different accelerators. The object replacement system must adapt to available computational resources, possibly switching between model variants of different sizes. Let $\theta^1, \theta^2, \ldots, \theta^K$ denote parameter sets for models of increasing capacity. A resource-aware controller selects a variant based on device characteristics and user settings, possibly trading off spatial resolution or temporal coverage for speed. The design of these variants involves pruning, quantization, and architectural simplification while attempting to keep behavior consistent across devices. Users should experience similar qualitative behavior even if quantitative fidelity varies.

7. Limitations and Ethical Considerations

Despite advances in modeling and system design, general-purpose object replacement in consumer video editors faces significant limitations. These limitations arise from the inherent ambiguity of visual content, constraints on computation and data, and the difficulty of capturing human expectations of realism and intent.

Ambiguity in object definition is a fundamental challenge. The notion of the "same object" across time is not always well defined, particularly when objects undergo occlusion, deformation, or appearance changes [35] [36]. For example, a person turning around may present different views with distinct textures, and partial occlusions by other objects may create uncertainty about object boundaries. Segmentation and tracking models rely on heuristics and learned priors to resolve these ambiguities, but their decisions may not align with user expectations. Mathematically, the space of plausible labelings M for a given video and user input may contain multiple local minima of the segmentation energy, and optimization routines may converge to different solutions depending on initialization and noise.

Generative synthesis introduces additional sources of uncertainty. When specification s is underdetermined, such as a short text description without precise attributes, the conditional distribution $p_\theta V' \mid V, s$ may place significant probability mass on diverse outcomes. Sampling from this distribution or selecting a single mode can yield edits that satisfy the specification in a broad sense but differ from what the user had in mind. While exposing controls to guide sampling can help, many users may not wish to adjust numerous parameters. From a statistical perspective, the posterior over edits may be multimodal, and presenting a small number of samples cannot fully represent its variability.

Computational limitations constrain model complexity and training data scale. Consumer-grade hardware limits the depth and width of networks that can be deployed, as well as the resolution and duration of videos that can be processed interactively [37]. This may force models to rely on aggressive downsampling, reduced temporal context, or simplified architectures, which can degrade performance on challenging content such as fast motion, fine textures, or long-range occlusions. In addition, training such models requires large datasets of videos that cover diverse environments, objects, and motions. If training data are biased toward specific settings, the learned priors may generalize poorly to underrepresented domains, leading to systematic artifacts or failures in certain types of content.

Ethical and societal considerations arise because object replacement can be applied not only for benign editing tasks

but also for deceptive manipulation. The ability to substitute objects in video with high visual fidelity can be misused to create misleading clips that distort events or attribute actions to entities that were not present. While consumer editors have long supported compositing and visual effects, the increasing realism and accessibility of generative methods lower the technical barrier for producing convincing manipulations. The risk is not limited to deliberate misuse; casual users may unintentionally create content that others interpret as factual.

Mitigation strategies depend on both technical and policy decisions. On the technical side, systems can incorporate provenance tracking, such as maintaining metadata that records the application of object replacement and other generative effects to a clip. Watermarking techniques that embed imperceptible signals indicating that a clip has been edited can also contribute, although their robustness and adoption may vary. Interfaces can display clear indicators when previews or exports contain synthetic content generated by object replacement, helping users understand the nature of the edits they are performing. On the policy side, guidance and terms of use can discourage harmful applications and outline responsibilities for users and platforms.

Another limitation concerns interpretability and control. The internal representations used by generative models and segmentation networks are often high dimensional and opaque [38]. Users may find it difficult to predict how small changes in input or parameters will affect the outcome. When the system makes unexpected decisions, such as subtly altering background elements or changing object attributes that the user intended to keep, it can be challenging to diagnose the cause or correct the behavior. Future work could explore more interpretable interfaces that expose meaningful intermediate structures, such as simplified geometry or semantic attributes, without overwhelming users with complexity.

Finally, object replacement raises questions about authorship and originality. When generative components synthesize new visual content conditioned on existing videos and user inputs, the resulting clips may blend contributions from dataset priors, user intent, and model design. Determining attribution for such content can be complex. While these questions extend beyond the scope of any specific technical system, they are part of the broader context in which general-purpose object replacement is developed and deployed.

8. Conclusion

This paper has examined general-purpose object replacement in consumer video editors from the perspectives of problem formulation, representation, algorithm design, numerical considerations, system integration, and limitations. The central view is that object replacement can be modeled as a constrained generative editing problem operating on layered spatiotemporal representations of video. A combination of masks, feature tensors, motion fields, and auxiliary depth or layer information provides a structured substrate on which generative operators can act, enabling the substitution of object appearance while preserving background content and enforcing temporal coherence.

Mathematically, the problem admits descriptions in terms of linear operators for compositing, probabilistic models for conditional generation, and graph-based energies for segmentation and tracking. These formulations highlight the interactions between discrete and continuous aspects of the task, including the discretization of time and space, the treatment of uncertainty, and the use of multivariate calculus to guide optimization. Regularization terms enforcing spatial and temporal smoothness, along with priors on motion and depth, help stabilize the editing process and reduce artifacts such as flicker or inconsistent occlusions [39].

From a systems perspective, integrating object replacement into consumer editors requires attention to computational budgets, interaction latency, and interoperability with other effects. Feature downsampling, tiling, caching, and model variants tailored to different hardware profiles can make advanced editing capabilities accessible on a wide range of devices. User interfaces must balance simplicity and control, allowing users to specify objects and replacements with minimal effort while still offering options to refine results or resolve ambiguities. Mechanisms for visualizing masks, uncertainty, and intermediate representations can support this goal without exposing the full complexity of the underlying models.

The discussion of limitations and ethical considerations underscores that object replacement is not purely a technical problem. Ambiguity in object definitions, variability in user intent, data and model biases, and the potential for misuse all influence how such systems should be designed and deployed. Technical mitigations such as provenance tracking, watermarks, and clear user interface cues can contribute to responsible use, but they must be complemented by broader norms and policies around synthetic media.

Future developments may extend the techniques surveyed here in several directions. More explicit three-dimensional scene representations, combined with efficient rendering and learning algorithms, could improve geometric and lighting consistency without prohibitive computational cost. Advances in uncertainty modeling and interactive learning might allow systems to adapt to individual user preferences and specific content domains more effectively, reducing the need for manual corrections. Improvements in interpretability and user-centric design could make advanced controls more approachable, helping users understand and guide complex edits. At the same time, ongoing reflection on the social and ethical implications of powerful editing tools will remain important as capabilities continue to grow. General-purpose object replacement in consumer video editors lies at the intersection of computer vision, graphics, machine learning, numerical analysis, and human-computer interaction. The formulations and considerations presented here aim to clarify the technical structure of the problem and its practical constraints, providing a foundation for further work on methods that are accurate, efficient, controllable, and aligned with responsible use in everyday editing workflows [40].

Acknowledgments

We would like to thank the reviewers of this research for their helpful comments and suggestions.

References

[1] G. Glowacki, A. Gkantiragas, B.-H. B, P. He, and M. D, "Development of an open source program to identify the size of objects using bright-field microscopy - an automated calibration tool," 5 2021.

[2] S. U. Yusuf and A. A. Fuad, "Real time implementation of face recognition based smart attendance system," *WSEAS TRANSACTIONS ON SIGNAL PROCESSING*, vol. 17, pp. 46–56, 7 2021.

[3] M. Mejia-Herrera, J. S. Botero-Valencia, D. Betancur-Vásquez, and E. A. Moncada-Acevedo, "Low-cost system for analysis pedestrian flow from an aerial view using near-infrared, microwave, and temperature sensors.," *HardwareX*, vol. 13, pp. e00403–e00403, 2 2023.

[4] L. Ali, Z. He, W. Cao, H. T. Rauf, Y. Imrana, and B. B. Heyat, "Mmdd-ensemble: A multimodal data-driven ensemble approach for parkinson's disease detection.," *Frontiers in neuroscience*, vol. 15, pp. 754058–, 11 2021.

[5] M. B. Altman, W. Wan, A. S. Hosseini, S. A. Nowdeh, and M. Alizadeh, "Machine learning algorithms for fpga implementation in biomedical engineering applications: A review.," *Heliyon*, vol. 10, pp. e26652–e26652, 2 2024.

[6] S. S. Harsha, D. Agarwal, A. Revanur, and S. Agrawal, "Digital video editing based on a target digital image," Aug. 21 2025. US Patent App. 18/583,067.

[7] A. Kalboussi, N. Omheni, O. Mazhoud, and A. H. Kacem, *INTERACT (2) - How to Organize the Annotation Systems in Human-Computer Environment: Study, Classification and Observations*, pp. 115–133. Germany: Springer International Publishing, 8 2015.

[8] K. Alsbury-Nealy, H. Wang, C. Howarth, A. Gordienko, M. L. Schlichting, and K. Duncan, "Openmaze: An open-source toolbox for creating virtual navigation experiments," *Behavior research methods*, vol. 54, pp. 1–14, 9 2021.

[9] B. Yasmeen, H. Arshad, and H. Rahman, "Suspicious activity detection using cctv surveillance video," *Journal of Information System and Technology Management*, vol. 6, pp. 60–70, 9 2021.

[10] N. Shakhovska and B. Kalysh, "The ensemble of neural networks to track eye movements," in *2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT)*, pp. 1–7, IEEE, 10 2023.

[11] M. Terras, "Another suitcase, another student hall — where are we going to? what ach/allc 2001 can tell us about the current direction of humanities computing," *Literary and Linguistic Computing*, vol. 16, pp. 485–491, 11 2001.

[12] T. Karsenti and S. Collin, "The impact of online teaching videos on canadian pre-service teachers," *Campus-Wide Information Systems*, vol. 28, pp. 195–204, 6 2011.

[13] C. Herrera, R. Baker, J. Sheaves, and M. Sheaves, "A computer vision approach for studying fossorial and cryptic crabs," 5 2020.

[14] K. A. Ritter and T. L. Chambers, "Educational gaming and use for explaining alternative energy technologies," *International Journal for Innovation Education and Research*, vol. 2, pp. 30–42, 3 2014.

[15] S.-M. Hu and L. Kobbelt, "Preface," *Journal of Computer Science and Technology*, vol. 30, pp. 437–438, 5 2015.

[16] Q. He and S. Fang, "Phantom-csi attacks against wireless liveness detection," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, vol. 1, pp. 440–454, ACM, 10 2023.

[17] S. Toral, F. Barrero, M. del Rocío Martínez-Torres, and S. Gallardo, "Interactive multimedia teaching of digital signal processors," *Computer Applications in Engineering Education*, vol. 15, pp. 88–98, 4 2007.

[18] F. Gmeiner, H. Yang, L. Yao, K. Holstein, and N. Martelaro, "Exploring challenges and opportunities to support designers in learning to co-create with ai-based manufacturing design tools," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–20, ACM, 4 2023.

[19] C. Han, M. Wang, and A. Yan, "The teaching method of basketball training based on intelligent network multimedia technology," *Mobile Information Systems*, vol. 2021, pp. 1–7, 10 2021.

[20] F. Battisti, M. Carli, F. Colangelo, and A. Neri, "A multi-objective optimization for video orchestration," 1 2018.

[21] M. Asif and Z. Gouqing, "Innovative application of artificial intelligence in a multi-dimensional communication research analysis: a critical review," *Discover Artificial Intelligence*, vol. 4, 5 2024.

[22] S. Purawat, C. Cowart, R. E. Amaro, and I. Altintas, "Biomedical big data training collaborative (bbdtc): An effort to bridge the talent gap in biomedical science and research.," *Journal of computational science*, vol. 20, pp. 205–214, 3 2017.

[23] H. Sun, W. Ni, and P. Zhao, "Design of a media resource management system for colleges based on cloud service," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, pp. 231–245, 11 2020.

[24] M. Wald and K. Bain, "Universal access to communication and learning: the role of automatic speech recognition," *Universal Access in the Information Society*, vol. 6, pp. 435–447, 10 2007.

[25] T. Mahalingam and M. Subramoniam, "Optimal object detection and tracking in occluded video using dnn and gravitational search algorithm," *Soft Computing*, vol. 24, pp. 18301–18320, 11 2020.

[26] Y. Li, L. Hu, K. Xia, and J. Luo, "Fast distributed video deduplication via locality-sensitive hashing with similarity ranking," *EURASIP Journal on Image and Video Processing*, vol. 2019, pp. 1–11, 3 2019.

[27] F. Filicori, D. P. Bitner, H. F. Fuchs, M. Anvari, G. Sankaranaryanan, M. B. Bloom, D. A. Hashimoto, A. Madani, P. Mascagni, C. M. Schlachta, M. Talamini, and O. R. Meireles, "Sages video acquisition framework-analysis of available or recording technologies by the sages ai task force.," *Surgical endoscopy*, vol. 37, pp. 4321–4327, 2 2023.

[28] J. Martin, D. Cantero, M. González, A. Cabrera, M. Larrañaga, E. Maltezos, P. Lioupis, D. Kosyvas, L. Karagiannidis, E. Ouzounoglou, and A. Amditis, "Embedded vision intelligence for the safety of smart cities.," *Journal of imaging*, vol. 8, pp. 326–326, 12 2022.

[29] U. Sulubacak, O. Caglayan, S.-A. Grönroos, A. Rouhe, D. Elliott, L. Specia, and J. Tiedemann, "Multimodal machine translation through visuals and speech," *Machine Translation*, vol. 34, pp. 97–147, 8 2020.

[30] A. Biglari and W. Tang, "A review of embedded machine learning based on hardware, application, and sensing scheme.," *Sensors (Basel, Switzerland)*, vol. 23, pp. 2131–2131, 2 2023.

[31] S. Mansour, S. B. Jabra, and E. Zagrouba, "A comprehensive overview of deep learning based video watermarking: current works, challenges and future trends," *Multimedia Tools and Applications*, vol. 84, pp. 28013–28060, 9 2024.

[32] S. S. A. Ghadani* and D. C. Jayakumari, "Innovating fire detection system fire using artificial intelligence by image processing," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, pp. 349–356, 9 2020.

[33] A. Benedek, "The imagistic turn in education: Opportunities and constraints," in *Proceedings of the International and Interdisciplinary Conference IMMAGINI? Brixen, Italy, 27–28 November 2017.*, vol. 1, pp. 855–, MDPI, 11 2017.

[34] C. Tan, G. Ceballos, N. Kasabov, and N. P. Subramaniyam, "Fusionsense: Emotion classification using feature fusion of multimodal data and deep learning in a brain-inspired spiking neural network.," *Sensors (Basel, Switzerland)*, vol. 20, pp. 5328–, 9 2020.

[35] R. Bao and J. Chen, "Everyone matters: A multimodal learning analysis framework based on individual time series," *Technology, Knowledge and Learning*, vol. 30, pp. 1269–1298, 5 2024.

[36] S. S. Harsha, A. Revanur, D. Agarwal, and S. Agrawal, "Genvideo: One-shot target-image and shape aware video editing using t2i diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7559–7568, 2024.

[37] X. Duan, Q. Wu, C. Gan, Y. Zhang, W. Huang, A. van den Hengel, and W. Zhu, "Acm multimedia - watch, reason and code: Learning to represent videos using program," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 1543–1551, ACM, 10 2019.

[38] P. Fraternali, F. Milani, R. N. Torres, and N. Zangrando, "Black-box error diagnosis in deep neural networks for computer vision: a survey of tools," *Neural Computing and Applications*, vol. 35, pp. 3041–3062, 12 2022.

[39] M. Sholeh, S. Suraya, and D. Andayani, "Learning at home during the covid-19 pandemic by optimizing video based tutorials," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 5, pp. 206–217, 8 2021.

[40] E. Krsak and S. Toth, "Traffic sign recognition and localization for databases of traffic signs," *Acta Electrotechnica et Informatica*, vol. 11, 1 2011.