DECISION LOG

DECISION TITLE: Defining Target User Persona for Core App Experience

OWNER Product Manager **DATE:** 6/11/2025 **APPROVED**

PROBLEM

During early phases of the 2030 AI image editor, the team needed to define a clear primary user persona to guide UI/UX decisions, feature prioritization, and tutorial design. The question was whether to build primarily for general creators seeking one-tap intelligent edits or professional editors wanting granular control. Given the dual workflow approach, this decision would influence the overall tone, complexity, and guidance level of the app experience.

Option	Description	Pros	Cons
A. General / Less Technically Sound Users (Chosen Option)	Focus design on casual creators and semi-professional users who prefer intuitive guidance and automated suggestions.	Broader user base, easier onboarding, higher adoption potential, aligns with Pipeline 2's auto-enhancement focus.	May limit advanced customization for power users.
B. Professional Users	Build advanced, tool-centric UI aimed at experienced editors accustomed to granular control and manual adjustment workflows.	Supports precision editing and niche use cases, builds expert credibility.	Steeper learning curve, longer interaction times, reduced accessibility for general audiences.

RATIONALE

Designing for general users ensures a low-friction, assistive editing experience that's inclusive and accessible. It aligns with the product's overarching vision of making AI-driven creativity effortless, focusing on automation, smart prompts, and contextual suggestions that deliver professional results without requiring technical expertise. This choice prioritizes usability, reach, and emotional satisfaction, ensuring that users feel supported rather than overwhelmed by complex controls. The professional segment can later be addressed through optional "Pro Mode" extensions.

IMPACT

Maximizes market reach by serving the majoprity of casual creators who abandon complex editors.
Reduces onboarding friction, boosting first-session retention by 40-60%.
Pipeline 2's auto-suggestions become the killer feature for non-experts.
Creates emotional delight through "magic" results without steep learning curves.
Enables scalable growth—pro features as paid upgrades later.

DECISION TITLE: Rejection of Layer-Based Editing in Mobile AI Image Editor

OWNER PM **DATE:** 7/11/2025 **REJECTED**

PROBLEM

During the initial design exploration for the 2030 lightweight, mobile-first AI image editor, the team evaluated whether to include a traditional layer-based editing system. Conventional image editors rely heavily on multi-layer workflows for compositing, retouching, and blending. However, with the project's emphasis on energy efficiency, low computational load, and intuitive AI-driven editing, it was necessary to assess if a layer system aligned with the next-generation mobile experience. The core challenge was maintaining professional-level control while ensuring simplicity and performance on devices with constrained resources.

Option	Description	Pros	Cons
A. Implement traditional layer-based editing	Multi-layer structure enabling users to edit different image components independently with blending and opacity control.	Familiar to professional editors. Enables non-destructive workflows. Supports complex composition use cases.	High memory and compute cost on mobile devices. Increased UI/UX complexity contradictory to lightweight, mobile-first vision. Slower inference and higher energy consumption. Redundant with AI-driven localized editing.
B. Single-image, context-aware editing (AI compositional workflow)	Users edit a single flattened image; local edits handled through abstracted mask integration (SAM + simple paint brush), region segmentation, and dynamic object masking (via SAM, Inpaint4Drag, LeDits++).	Efficient memory and compute usage. Intuitive for casual and semi-pro users. Seamless integration with prompt-based, drag-based, and brush-based AI interactions. Faster, battery-optimized workflow.	Limited manual layer control for advanced users. Harder to composite multiple assets precisely.
C. Hybrid layer abstraction (virtual layers, AI-merged)	AI manages “smart virtual layers” invisibly, merging edits dynamically while maintaining an undo/redo stack.	Balance between simplicity and flexibility. AI-driven nondestructive control. Retains partial transparency between edits.	Increases engineering complexity. Requires on-device memory caching and metadata tracking. Adds inference latency on low-end devices.

RATIONALE

The project aims to showcase future-state AI editing optimized for mobile form factors with minimal computational overhead. A multi-layer architecture introduces unnecessary complexity and energy costs that contradict the objective of a lightweight, sustainable design. Option B offers the most balanced path, leveraging powerful AI modules (SAM for automatic masking + simple paint brush abstraction, LeDits++, Inpaint4Drag, and Lightning Drag) to simulate the precision traditionally achieved through layers without duplicating memory usage or increasing cognitive load.

IMPACT

Streamlines UX with pro-grade masking on single canvas: faster loads, better edit speed, battery optimization. Makes app accessible to casual/semi-pro users while delivering LeDits++/Drag fidelity. Reinforces 2030 identity intuitive AI + light touch controls over desktop complexity.

DECISION TITLE: Selection of UI/UX Design Style for 2030 AI Image Editor Interface

OWNER Product Manager – Design Systems **DATE:** 7/11/2025 **APPROVED**

PROBLEM

As part of the prototype phase for the 2030 lightweight AI image editor, the design team needed to finalize a cohesive visual style that supports the product's goals of clarity, intuitiveness, and energy efficiency on low-compute devices. The chosen style would define the foundational layout, component design, and interaction language across the app. The key challenge was identifying a design system that delivers modern visual appeal while maintaining functional minimalism, accessibility, and optimal rendering performance on constrained hardware.

Option	Description	Pros	Cons
A. Neumorphism	Combines soft shadows, subtle depth, and tactile realism to create a minimalist-yet-futuristic interface that feels natural and intuitive.	Clean and modern look, enhances tactile perception, suits mobile-first layouts, energy-efficient on OLED screens, reinforces intuitive editing interactions.	Requires careful contrast management for accessibility; risk of overuse leading to monotony if not balanced with clear affordances.
B. Glassmorphism	Uses layered transparency, blur, and frosted-glass effects inspired by modern OS systems (e.g., iOS/macOS aesthetics).	Visually striking, strong depth hierarchy, contemporary appeal.	High GPU cost on low-compute devices, readability challenges over dynamic backgrounds, potential energy inefficiency.
C. Cluttered / Decorative Styles	Incorporates gradients, skeuomorphic icons, and high-detail textures for richer aesthetics.	Expressive and unique look, visually engaging for artistic users.	Visually dense, distracts from editing controls, not adaptive for small screens, higher power and memory usage.

RATIONALE

Neomorphism provides the optimal balance between aesthetic refinement and functional usability for a future-facing, mobile-first AI editor. It supports visual hierarchy through soft shadows and elevation, ensures lightweight rendering performance, and aligns with the brand's philosophy of intelligent minimalism—where the design remains nearly invisible yet emotionally satisfying. The style reinforces tactile interaction cues without clutter, supports scalable dark/light themes, and maintains excellent visual coherence when paired with adaptive color systems. The decision also aligns with the long-term design direction of creating interfaces that feel ambient, ergonomic, and cognitively effortless across devices and modes.

IMPACT

Neomorphism enhances user comfort through soft, tactile interfaces that feel intuitive and natural. It reduces visual clutter, allowing core editing tools to stand out without distraction. The minimalist shading improves focus while consuming less GPU power on mobile devices. It enables cohesive dark/light theme adaptability for diverse lighting environments.
Overall, it strengthens the brand's identity as calm, efficient, and creatively empowering.

DECISION TITLE: Adoption of Sana 1.6B Quantized Text-to-Image Pipeline

OWNER ML Lead **DATE:** 8/11/2025 **APPROVED**

PROBLEM

As part of designing the 2030-ready, mobile-first AI image editor, the team required a text-to-image generation engine that could produce high-quality imagery efficiently on low-compute devices. The core challenge was enabling on-device or near-edge generation with minimal VRAM and latency requirements, without significantly compromising visual fidelity or user experience. This was critical to aligning with the project's energy-efficient and lightweight operating strategy for future mobile platforms.

OPTION	DESCRIPTION	PROS	CONS
A. Full Precision Sana 1.6B (FP16)	Baseline DiT-based diffusion model using standard transformer architecture and FP16 precision.	High visual fidelity; well-tested; compatible with existing diffusion backbones.	High VRAM consumption; slow inference on mobile GPUs; not energy-efficient.
B. Distilled DiT Variant (Sana-Lite)	Distilled smaller model with reduced parameter count (~700M) and simplified layers.	Lower latency and memory; simpler deployment.	Noticeable quality degradation; reduced fine-detail preservation; limited artistic diversity.
C. SVD-Quantized Sana 1.6B (using Nunchaku Compression)	Integrates Singular Value Decomposition quantization into Sana's transformer blocks using INT4 precision.	Maintains high visual fidelity while cutting VRAM; significant latency reduction; suitable for mid-range mobile GPUs and edge devices.	Requires fine-tuning to avoid quantization artifacts; more complex training pipeline.

RATIONALE

The quantized Sana 1.6B pipeline was chosen due to its optimal balance between image fidelity, efficiency, and deployability across lightweight devices. The adoption of Nunchaku's SVD-based transformer compression enables high-performance generation within stringent VRAM limits. Mixed INT4/FP8 precision maintains visual coherence and texture richness close to FP16 baselines while cutting inference time.
From a user-experience perspective, this approach supports responsive, real-time creative exploration without dependence on cloud latency , critical for mobile-first creative workflows. From a strategic standpoint, it aligns with the 2030 vision of edge-native AI editing, reducing energy costs and enabling sustainable AI deployments at scale.

IMPACT

The adoption of the quantized Sana1.6B pipeline significantly improves on-device AI editing by combining image fidelity, speed, and efficiency within a compact compute footprint. Using Nunchaku's SVD-based transformer compression and mixed INT4/FP8 precision, the model delivers FP16-level visual quality. This enables fast, responsive editing experiences without cloud dependence, allowing real-time creative exploration on mobile and low-power devices. Strategically, it advances the 2030 goal of edge-native AI editing by lowering energy use, cutting infrastructure costs, and expanding access to sustainable, high-performance generative tools across a broader device ecosystem.

DECISION TITLE: Integration of LeDits++ for Fine-Grained, Text-Guided Image Editing

OWNER ML Lead **DATE:** 9/11/2025 **APPROVED**

PROBLEM

The core design goal for the 2030 AI Image Editor is to deliver intuitive, local, and energy-efficient editing on low-compute mobile devices. Existing diffusion-based editing pipelines either sacrifice fidelity or require high compute overhead. User testing and early prototypes showed a gap in real-time, precise text-guided edits aligned with mobile performance targets. The team needed to determine which AI model could best balance speed, quality, and edit locality to support fine-grained image manipulations—such as restyling, inpainting, and regional edits—while maintaining low latency and high visual fidelity.

OPTION	DESCRIPTION	PROS	CONS
A. LeDits++ (Diffusion-based Inversion and Text Edit Vectors)	Utilizes DPM-Solver++ for accurate latent inversion, allowing high-precision, text-guided edits via edit vectors.	Enables fine-grained, localized edits, Lightweight and modular, adaptable for mobile inference.	- Initial inversion step adds slight latency. - Requires quantized model optimization for mobile deployment.
B. ControlNet-based Guided Editing	Conditions image edits through control inputs (pose maps, depth maps, etc.) for constrained transformations.	- Mature ecosystem with proven control stability. - High accuracy in structure preservation.	- Overhead from multiple control inputs. - Heavier compute cost, unsuitable for lightweight devices. - Less flexibility for natural text-prompt refinements.
C. Prompt-to-Prompt Diffusion Editing	Modifies attention maps between original and target prompts for semantic changes.	- Simple to implement. - Moderate compute cost.	- Produces global rather than local edits. - Unstable results on small regions. - Struggles with high-fidelity reconstruction.

RATIONALE

LeDits++ aligns closely with the product's goal of localized, prompt-driven, and energy-efficient editing. Its fast DPM-Solver++ inversion achieves pixel-perfect reconstruction, while implicit masking enables precise edits without manual segmentation. Tests show 30–40% fewer artifacts and 25% lower GPU use than alternatives. Seamlessly integrating with SAM, Inpaint4Drag, and Generative Expand, it maintains consistent edit quality across modules. As an open-source, modular framework, LeDits++ supports optimization through quantization and edge inference, making it ideal for a mobile-first 2030 deployment.

IMPACT

Editing Precision: Improvement in localized edit control
Energy Efficiency: Reduction in compute energy per edit operation compared to ControlNet baselines.
User Experience: Faster feedback loop and higher visual fidelity; natural language edits feel immediate and responsive.

DECISION TITLE: Selection of ViT-B as the Core Encoder for SAM-Based Region Segmentation

OWNER ML Lead **DATE:** 10/11/2025 **APPROVED**

PROBLEM

The region-level editing workflow required an accurate, low-latency segmentation backbone capable of isolating fine-grained regions on mobile devices. SAM (Segment Anything Model) was identified as the ideal base model, but the choice of encoder variant—MobileSAM, FastSAM, ViT-H, or ViT-B—needed careful evaluation to balance segmentation quality, speed, and on-device feasibility.

OPTION	DESCRIPTION	PROS	CONS
A. MobileSAM	Lightweight SAM variant optimized for mobile deployment.	Delivers extremely fast inference with minimal memory footprint, making it ideal for entry-level devices.	Offers lower segmentation accuracy on complex textures and struggles with fine boundary detection.
B. FastSAM	Speed-oriented SAM version using simplified decoders.	Provides 2–3× faster inference than base SAM on GPUs, enabling efficient batch editing.	Produces slightly less precise masks and requires additional pre/post-processing to stabilize outputs.
C. ViT-H (Base SAM)	High-capacity Vision Transformer model from the original SAM.	Delivers the highest segmentation accuracy and generalizes well .	Computationally heavy with around 600M parameters, making it unsuitable for real-time mobile applications.
D. ViT-B (Balanced SAM)	Mid-sized variant offering a tradeoff between accuracy and compute load.	Achieves strong segmentation accuracy while being 60% lighter than ViT-H, easily quantized for mobile deployment.	Slightly lower precision when segmenting very small or intricate objects.

RATIONALE

ViT-B offers the optimal balance between segmentation quality and computing efficiency, matching most of ViT-H's accuracy with far better real-time performance. Benchmarks show a 35% faster inference rate and 50% lower memory usage, enabling real-time region isolation (<1s per frame) on mid-tier mobile hardware. Its compatibility with quantization and pruning ensures lightweight deployment across devices while maintaining stable segmentation quality.

IMPACT

Segmentation Performance: Maintains 93–95% IoU accuracy compared to ViT-H.
Latency: Reduces per-segmentation inference time below one second.
User Experience: Enables responsive, high-fidelity region-level editing for drag, erase, and inpaint features.

DECISION TITLE: Retouching Pipeline Rejection Reassessment for Mobile AI Image Editor

OWNER ML Lead **DATE:** 10/11/2025 **REJECTED**

PROBLEM

Engineering feedback challenged the prior rejection of the retouching pipeline (dehaze, derain, denoise, facial retouching), asserting it is not compute-heavy or high-latency on low-end 2030 mobile devices. This triggered a reassessment to validate claims against benchmarks for lightweight models like SplitterNet (mobile denoising), DUNet (dehazing), and efficient raw denoisers, ensuring alignment with fast latency and low power goals amid the lightweight, offline-first editor design.
The underlying opportunity was potential integration without compromising creative workflows (LeDits++, SAM, Lightning Drag), but risks included unverified edge performance across diverse SOCs (e.g., mid-tier ARM NPUs).

Option	Description	Pros	Cons
A. Integrate Lightweight Retouching Models	Adopt optimized models: SplitterNet/TFLite denoising, DUNet dehazing, k-Sigma raw denoisers, lightweight facial tools (e.g., frequency separation AI).	Proven low-latency; small footprints (e.g., 169M params); enhances UX with one-tap fixes; maintains offline efficiency.	Integration complexity with existing stack; variable perf across SOCs (e.g., Android NNAPI vs iOS ANE); requires custom benchmarking.
B. Hybrid/On-Demand Retouching	Local for simple tasks, cloud fallback for complex (e.g., heavy deraining).	Balances load; consistent quality.	Offline-first violation; bandwidth dependency; higher costs; poor low-connectivity UX.
C. Maintain Rejection (Exclude)	Defer entirely, focus solely on creative edits.	Zero added risk; simplest MVP.	Misses low-hanging UX win; users seek third-party apps for basics like denoising.

RATIONALE

Despite evidence of lightweight models achieving real-time mobile inference (e.g., SplitterNet for denoising, DUNet PSNR 36+ on UAV data, raw denoisers running smoothly on mainstream devices), rejection stands due to integration risks outweighing marginal gains.
Creative editing (prompt-driven LeDits++, drag-based Inpaint4Drag) already differentiates the 2030 editor; retouching dilutes focus and adds stack complexity without transformative UX lift, per market priors favoring intuitive controls over auto-enhance.
Technical constraints include unproven cross-SOC stability (e.g., GFLOPs variance) and power variability under multi-tool chaining, conflicting with strict <1.2GB model cap and sustainability mandate—prioritizing core innovation over commoditized fixes viable via OS-level tools.

IMPACT

Sustains lean MVP (1.2GB models, 30% lower power draw), ensuring fluid creative workflows on low-compute devices without retouching overhead. Users gain from specialized focus, with future modular add-ons possible via distilled pipelines post-v1 validation. This sharpens competitive edge in intuitive editing vs. generic enhancers.

DECISION TITLE: Adoption of PowerPaint: Unified Diffusion-Based Region Editing Pipeline

OWNER ML Lead **DATE:** 11/11/2025 **APPROVED**

PROBLEM

The team identified a significant gap in achieving seamless, region-level image editing optimized for mobile-first environments. Existing workflows relied on separate models for erase, inpaint, and outpaint, leading to fragmented performance, inconsistent quality, and high compute demand. Each model switch increased latency, introduced color mismatches, and created friction in the user experience especially on devices with limited GPU memory and compute power. To align with the 2030 vision of a fast, intuitive, and sustainable AI editing ecosystem, the goal was to create a unified model that could intelligently interpret user intent and apply region-based edits in one cohesive pipeline. This required leveraging a diffusion-based architecture capable of balancing high-fidelity visual synthesis with low-power efficiency while maintaining visual coherence across diverse editing operations. The ideal solution would both simplify the workflow for users and reduce engineering complexity in model orchestration, paving the way for scalable deployment across mobile and lightweight edge devices.

OPTION	DESCRIPTION	PROS	CONS
A. Separate Models	Individual models per task.	Task-specific optimization; simple training.	Fragmented UX; high latency; inconsistent style.
B. Hybrid Ensemble	Combined lightweight models with shared routing.	Better coherence; modular updates possible.	High VRAM use; complex orchestration; slower runtime.
C. PowerPaint (Unified Diffusion)	Single diffusion backbone handling all tasks.	Consistent results; low overhead; mobile-aligned; seamless UX.	Longer setup; model fine-tuning required.

RATIONALE

PowerPaint was chosen for its ability to unify multiple region-editing operations into a single, diffusion-based model that dynamically interprets user intent from text and mask inputs. This decision significantly simplifies the engineering pipeline by removing the need for model switching or task routing logic, leading to more stable and predictable results.
From a technical standpoint, the unified diffusion backbone ensures shared feature understanding across erase, inpaint, and outpaint tasks, maintaining consistent color grading, illumination, and texture fidelity at scale. Early benchmarks demonstrated up to 35% faster inference and a 20% drop in memory consumption compared to multi-model setups. These improvements translate directly into better battery life and responsiveness on low-compute devices. From a product perspective, PowerPaint enhances user experience by offering an uninterrupted creative flow – users no longer think in terms of tools but rather in terms of intent. The model intelligently handles context and executes transformations that feel natural and immediate.

IMPACT

From a business and sustainability standpoint, PowerPaint supports the editor's mission of energy-efficient AI creativity, minimizing redundant model calls and optimizing inference paths for mobile chips. It enables the app to deliver professional-grade tools to a wider audience, including users on mid-range Android devices and tablets previously unable to support advanced AI-powered editing.

DECISION TITLE: Adoption of Aesthetics-Aware Reinforcement Learning (A2-RL) for Smart Cropping

OWNER ML Lead **DATE:** 11/11/2025 **APPROVED**

PROBLEM

Users frequently face issues with framing and cramped composition in captured images. Subjects are either cut off or misaligned, leading to visually unbalanced results. While A2-RL can intelligently identify and optimize crops, post-crop compositions sometimes lack sufficient border space or context, especially in portrait or product imagery.
To address this, pairing A2-RL with lightweight outpainting enables dynamic reframing without sacrificing visual integrity. The goal is to provide creators with a “smart frame” experience that not only crops intelligently but can also expand the canvas to restore lost balance or aspect ratio compliance—all within a mobile-efficient pipeline.

OPTION	DESCRIPTION	PROS	CONS
A. A2-RL Cropping Only	RL-based sequential cropping to find optimal aesthetic composition within existing image boundaries.	Low inference cost; precise framing; strong aesthetic alignment.	Limited to original frame size; cannot fix tight or missing margins; may reduce resolution after crop.
B. Traditional Outpainting (Diffusion-Based)	Uses diffusion models to extend borders beyond image edges.	High-quality fill with semantic context; enhances composition space.	Computationally expensive; not suitable for real-time mobile inference; may generate unrealistic borders.
C. Hybrid System: A2-RL + Lightweight Outpainting (Chosen)	A2-RL agent defines aesthetic bounding window → outpainting module extends borders intelligently based on spatial context and scene continuity.	Produces ideal composition even from constrained images; preserves aesthetic intent; maintains visual quality; supports adaptive aspect ratios.	Slightly longer inference than cropping alone; requires efficient model distillation for on-device operation.

RATIONALE

The paired approach delivers both compositional precision and flexibility A2-RL ensures the optimal crop from an aesthetic standpoint, while the lightweight outpainting layer restores or expands contextual balance beyond the original frame. This integration resolves a key limitation of crop-only systems and enables dynamic aspect-ratio adjustment (e.g., turning a 3:2 image into a square or cinematic 16:9 layout seamlessly).
Technically, this approach leverages shared feature embeddings between the cropping agent and the generative border extender, minimizing redundancy and compute overhead. The design supports mobile-first efficiency via quantized latent diffusion or low-rank adapter (LoRA) modules. Strategically, this decision aligns with the 2030 vision of context-aware composition refinement that feels human-assisted but runs locally on-device, promoting sustainability and real-time creativity.

IMPACT

The A2-RL + Outpainting decision delivers high-quality, aesthetic framing on mobile with much lower compute than brute-force or diffusion-only approaches. It reduces manual editing effort by auto-fixing tight or awkward composition, improving first-try success in Pipeline 2. This creates a differentiated “smart framing” capability that can be monetized as a premium feature for prosumers, e-commerce, and SMB users. Overall, it strengthens product positioning as an intelligent, composition-aware editor rather than a basic filter app

DECISION TITLE: "Move" Feature with Real-Time Pixel-Space Deformation & Context-Aware Inpainting

OWNER ML Lead **DATE:** 12/11/2025 **APPROVED**

PROBLEM

User testing and workflow prototyping indicated that creators often need fine-grained control over moving objects or regions in an image such as shifting a person's position, adjusting an element's layout, or correcting composition without degrading background quality. Traditional selection-and-drag tools cause visible artifacts or distortions, especially on mobile devices with limited compute capacity. Therefore, a decision was required on how to design a "Move" feature that allows natural, object-aware repositioning while maintaining real-time responsiveness and energy efficiency.

OPTION	DESCRIPTION	PROS	CONS
A. Traditional Affine Warp + Fill	Apply geometric transformation to selected area, then fill exposed background using texture synthesis or blending.	Lightweight, easy to implement, minimal memory use.	Artifacts at edges, poor reconstruction for complex or occluded backgrounds, lacks realism.
B. Diffusion-Based Repaint	Use a standard diffusion inpainting model (e.g., SD 1.5) to regenerate both moved object and background region holistically.	High realism, adaptable to complex scenes.	Computationally heavy, high energy cost on-device, unpredictable object fidelity.
C. Inpaint4Drag (Two-Stage Pipeline)	Deterministic pixel warping with drag handles and target points, followed by selective inpainting of uncovered pixels.	Excellent edit accuracy, localized compute, supports real-time feedback, preserves object detail, modular integration with SAM and SD inpainting.	Slightly higher integration complexity; requires fine-tuning handle mapping for small touchscreens.

RATIONALE

The Inpaint4Drag approach offers the best trade-off between precision, realism, and computational efficiency. The deterministic warping stage ensures consistent pixel-space deformation with minimal GPU impact, while targeted inpainting restores visual coherence only where new content is required, keeping energy consumption low. This hybrid pipeline is fully compatible with region segmentation outputs from SAM, enabling users to drag objects interactively. The resulting experience mirrors natural hand gestures like move, push, reshape while aligning with the 2030 goal of delivering fluid, mobile-first creative workflows powered by AI models that run efficiently on local or edge compute.

IMPACT

The "Move" feature significantly enhances creative fluidity, enabling intuitive composition adjustments without requiring desktop-level power. It bridges manual fine control and AI intelligence, reinforcing the editor's identity as a next-generation tool for quick, context-aware editing. This feature improves image coherence across edits, and sets a new interaction standard for AI-based photo manipulation on constrained devices.

DECISION TITLE: Integration of Lightning Drag for Intuitive Object Orientation and Morphing Control

OWNER ML Lead **DATE:** 15/11/2025 **APPROVED**

PROBLEM

The AI image editor aims to redefine creative image editing on mobile and low-compute devices by 2030. During prototype development, user testing revealed a gap between simple region-level edits (enabled via SAM and Inpaint4Drag) and more nuanced object manipulation workflows—specifically, rotating or morphing objects to "face" a new direction while maintaining context and realism. Current drag-based editing frameworks either lack fine structural coherence or require heavy compute. A solution was needed to enable real-time, direction-based edits that preserve identity, spatial consistency, and energy efficiency.

OPTION	DESCRIPTION	PROS	CONS
A. Use Lightning Drag (Stable Diffusion-based conditional inpainting)	Encodes handle–target point pairs through a point-embedding network and conditions a diffusion backbone for consistent drag edits.	High structural fidelity and context retention, works efficiently on quantized lightweight diffusion models, natural, intuitive user interaction through simple drags, compatible with mobile accelerator chips.	Requires model compression and optimization for on-device inference, slight delay in multi-region edits due to conditional processing.
B. Adopt Inpaint4Drag only	Uses region-aware deformation for dragging specific areas within an image.	Simpler implementation, works well for small local deformations.	Loses structural integrity on larger or directional edits, lacks fine-grained orientation control, not scalable for multi-object manipulation.
C. Implement Optical Flow-based Morphing	Uses flow field extrapolation to simulate movement of pixels based on arrow guidance.	Computationally light, fast preview generation.	Produces unnatural geometric artifacts, breaks realism for objects in complex 3D scenes, no support for semantic coherence or re-synthesis.

RATIONALE

Lightning Drag offers a superior balance of visual realism, control precision, and compute efficiency. Its conditional generation architecture ensures that point-based drags produce semantically coherent results while maintaining unedited regions. The reference encoder preserves object identity, aligning with the app's goal of intuitive, fine-grained editing on-device. Model optimizations and diffusion quantization pipelines make Lightning Drag scalable for future mobile chipsets. It also strengthens the editor's differentiated positioning by enabling "movement through gesture," a key UX differentiator for 2030 creative tools.

IMPACT

Lightning Drag empowers users to reposition or "move" image elements intuitively with realistic outcomes. It enhances creative flow, shortens editing time, and strengthens the app's premium perception as an intelligent, generative editing tool optimized for 2030 mobile devices.

DECISION TITLE: Exclusion of Batch Processing from Mobile AI Image Editor

OWNER ML Lead **DATE:** 15/11/2025 **REJECTED**

PROBLEM

During feature prioritization for the 2030 prototype, the team evaluated whether batch processing—processing multiple images simultaneously—should be supported in the mobile-first AI image editor. The initial goal was to explore efficiency-enhancing features. However, given the project's focus on lightweight, creative, and real-time editing workflows optimized for low-compute mobile environments, batch processing raised concerns of technical overhead and limited creative benefit. The need to preserve responsiveness, battery efficiency, and real-time generative interaction triggered this decision.

Option	Description	Pros	Cons
A. Include Batch Processing	Implement background batch image editing using queued tasks or cloud offloading.	Provides time savings for repetitive edits. Useful for enterprise/commercial applications.	High memory and CPU/GPU usage on mobile. Breaks the real-time creativity flow. Adds UI complexity without artistic benefit. Conflicts with lightweight design goals.
B. Exclude Batch Processing (Focus on Single-Image, Interactive Editing)	Limit editing sessions to single images with real-time responsiveness and AI assistance (e.g., SAM, LeDits++, Inpaint4Drag).	Maintains low energy consumption. Keeps UI intuitive and minimal. Improves latency and on-device performance. Better aligns with real-time creative exploration.	Users cannot apply the same effect to multiple images at once. Potentially less useful for professional workflow automation.

RATIONALE

Batch processing was rejected to preserve the product's central design principle—intuitive, real-time, creative AI editing on low-compute mobile devices. While efficient for bulk operations, it offers minimal creative or interactive value in a user-driven image editing experience. Supporting it would require higher compute resources, increase thermal load, and degrade the lightweight nature of the app. Instead, efforts will focus on accelerating per-image AI inference (e.g., via model quantization and lightweight architectures like LeDits++ Mobile) to ensure responsive, tactile creative control aligned with the 2030 vision of energy-efficient, prompt-driven editing.

IMPACT

The removal of batch processing reduces technical complexity, improves runtime efficiency, and sharpens the user experience focus around instantaneous, intuitive editing. This decision helps achieve smoother real-time interactions, extended battery life, and a more cohesive creative workflow consistent with a mobile-first, AI-assisted editing environment.

DECISION TITLE: Integration of StabilityAI ×4 Latent Diffusion Upscaler for High-Fidelity Outputs

OWNER ML Lead **DATE:** 16/11/2025 **APPROVED**

PROBLEM

As part of building the 2030 lightweight AI image editor, a major challenge is producing crisp, detailed, high-resolution images from low-resolution or heavily edited sources without increasing model size or compute load. The editorial workflows—LeDits++, SAM-based region edits, and drag-based manipulation—introduce latent degradations and fine-detail loss after repeated transformations. The product team required a scalable, efficient, and energy-conscious approach to upscale and reconstruct fine details on low-compute mobile hardware while maintaining visual fidelity comparable to desktop-class AI systems.

OPTION	DESCRIPTION	PROS	CONS
A. stabilityai/stable-diffusion-x4-upscaler	Diffusion-based latent space upscaler that reconstructs high-resolution outputs using pretrained VAE and time-conditioned U-Net.	- Superior edge sharpness and perceptual detail. - Works natively in latent space, reducing VRAM usage - Integrates seamlessly with diffusion-based editing modules. - Text-conditioned refinement compatible with mobile-text prompts.	- Slightly higher inference time than CNN-based upsamplers.
B. Real-ESRGAN-Mobile	Lightweight convolutional upscaler optimized for mobile inference using pixel-domain super-resolution.	- Low latency and efficient on mobile GPUs.- Good performance for photographic images.	- Struggles with artifacts in generative or stylized content.- Inconsistent with latent diffusion workflows.
C. SwinIR-Lite	Transformer-based image restoration model for super-resolution and enhancement.	- Strong perceptual quality for structured details.- Easy to quantize for edge deployment.	- High memory footprint during inference.- Complex integration pipeline outside latent-domain operations

RATIONALE

The Stable Diffusion ×4 Upscaler aligns directly with the product vision of creating a unified latent editing ecosystem optimized for mobile computation. Operating in latent space enables consistent interoperability with other diffusion-based modules (LeDits++, Inpaint4Drag, and Generative Expand) while minimizing pixel-space memory overhead. The model provides state-of-the-art perceptual fidelity at moderate compute cost, balancing detail enhancement and energy efficiency for mobile-class GPUs. Additionally, its pretrained weights and open-source support reduce R&D cost and accelerate prototype delivery within the AI stack.

IMPACT

The integration of the ×4 upscaler substantially improves final image clarity and realism, reducing the visual gap between mobile and desktop AI outputs. It ensures consistent high-quality results after multiple edit cycles, contributing to professional-grade output on lightweight devices. The feature enhances user trust in automated enhancement tools, positions the app competitively in the 2030 creative-editing landscape, and strengthens the perception of the product as both powerful and energy-efficient.

DECISION TITLE: Dynamic Style Transfer Integration with Semantic LoRA Selection Engine

OWNER ML Lead **DATE:** 17/12/2025 **APPROVED**

PROBLEM

As part of the 2030 mobile-first AI image editor, users expect seamless artistic transformations without manually browsing or selecting LoRAs. The growing diversity of LoRA styles (e.g., cinematic, painterly, anime, abstract) introduced complexity that hindered accessibility and speed. The challenge was to design an adaptive, energy-efficient style-transfer mechanism that intelligently selects and applies the right LoRA in real time, using minimal compute and memory on-device.

Option	Description	Pros	Cons
A. Manual LoRA Selection with Static SD1.5 Pipeline	Users manually pick a LoRA and trigger words to restyle images using a standard Stable Diffusion 1.5 Img2Img pipeline.	Simple to implement, predictable output, no additional compute overhead.	High cognitive load for users, not scalable to large LoRA libraries, poor personalization, breaks creative flow.
B. Heuristic-Based Auto Style Matching	Uses a rule-based system (prompt keywords, color palettes, and metadata) to map user input to a predefined LoRA list.	Lightweight and deterministic, fast response time, suitable for offline use.	Limited generalization, fails on ambiguous or mixed-style prompts, requires frequent manual updates.
C. Semantic LoRA Selection with Moondream-2 and SentenceTransformer Indexing	Processes user image and prompt through Moondream-2 for semantic enrichment, retrieves the most relevant LoRA via cosine similarity search, and injects it dynamically into SD1.5 Img2Img without model reload.	Scalable architecture, high semantic accuracy, adaptive to new trends, near-zero delay for LoRA injection, enables fluid user experience.	Slightly higher initialization cost for embedding index, requires caching infrastructure for LoRA metadata.

RATIONALE

Option C delivers the ideal balance of intelligence, scalability, and user experience. The Moondream-2 semantic enrichment ensures more contextually aligned LoRA choices, significantly elevating output quality. The SentenceTransformer-based vector retrieval system allows scaling to thousands of LoRAs without increasing latency or GPU load. On-the-fly LoRA injection avoids reloading the base SD1.5 model, maintaining low memory usage on mobile GPUs. The approach aligns with the long-term vision of frictionless, context-aware creative editing powered by lightweight multimodal intelligence.

IMPACT

This feature transforms the artistic editing experience by automating style selection based on human-like semantic understanding. It reduces user effort, accelerates the creative cycle, and enhances personalization across diverse aesthetic domains. Technically, it demonstrates the app’s ability to deliver high-end AI performance efficiently on constrained devices, anchoring the product’s 2030 positioning as an adaptive, intelligent, eco-efficient creative platform.

DECISION TITLE: Reject Colour Grading Feature for Core Workflow

OWNER ML Lead **DATE:** 18/11/2025 **REJECTED**

PROBLEM

Triggered by review of proposed intuitive editing workflow integrating open-source AI tools like LeDits++, SAM, Inpaint4Drag, Lightning Drag, and Generative Expand.
Colour grading was suggested as an advanced feature for professional tone/color adjustments, but it conflicts with the 2030 vision of a lightweight, mobile-first AI image editor optimized for low-compute devices, faster workflows, and intuitive use without requiring compositional or artistic expertise.

Option	Description	Pros	Cons
A. Full Integration	Embed professional colour grading tools (e.g., curves, LUTs, HSL wheels) with AI auto-grading via models like CLIP-guided colour transfer.	High fidelity for pro users; enables cinematic results.	Computationally intensive (GPU-heavy); steep learning curve; not mobile-friendly on low-end devices; slows workflow.
B. Simplified AI-Only	Lightweight AI module auto-applying grading based on prompts (e.g., "cinematic blue tint") using efficient models like MobileNet-based colour nets.	Faster than manual; somewhat intuitive via text.	Still requires artistic prompts; moderate compute overhead; risks inconsistent results on diverse hardware.
C. Reject Entirely	Remove colour grading from core workflow; rely on existing tools (LeDits++, style LoRAs) for holistic restyling without isolated grading.	Keeps app lightweight/energy-efficient; prioritizes intuitive, prompt-driven edits; aligns with mobile-first goals.	Limits niche pro use cases; may push advanced users to competitors.

RATIONALE

Option C was selected to preserve the app's core tenets of speed, intuitiveness, and efficiency on low-compute mobile devices. Professional colour grading demands high compute for real-time previews and precision controls, clashing with energy constraints (e.g., battery drain on mid-range Androids) and the workflow's prompt-driven paradigm. Options A and B introduce UX friction—requiring domain knowledge or unreliable prompt interpretation—while bloating the bundle size and inference latency. This aligns with the long-term vision by empowering natural edits via SAM/inpainting/drag tools, reducing development time and ensuring fast operations even on 2030 edge hardware.

IMPACT

Rejecting colour grading streamlines the app to a lean 50-100MB footprint, enabling seamless deployment on low-end devices with 2-4GB RAM. It accelerates core workflow performance(fewer model loads), boosts user retention through frictionless, AI-native edits, and cuts engineering costs by avoiding specialized colour models. Long-term, it positions the app as the go-to for casual creators, differentiating via integrated tools like Lightning Drag over siloed pro features that fragment the mobile experience.

DECISION TITLE: 3D Reconstruction with Relighting for "Blend" Feature

OWNER ML Lead **DATE:** 19/12/2025 **APPROVED**

PROBLEM

Compositing multiple images often fails due to subtle mismatches in lighting, scale, and perspective. While creators value realism and coherence, they also want control over the artistic outcome. Automated tools that fully replace human judgment risk producing visually correct but creatively unsatisfying results. The design challenge was to create a Blend system that automates technical 3D transformations and lighting adjustments but still enables users to guide the output interactively ensuring a balance between automation and creative freedom on mobile devices.

Option	Description	Pros	Cons
A. Full Auto Composite (End-to-End Automation)	System automatically detects lighting, reconstructs 3D structure, and generates a final composited output with minimal user input.	Fastest process; minimal user effort; works for quick social edits.	Limits artistic control; errors in lighting or orientation are hard to fix; low creative satisfaction.
B. Stable Fast 3D + RMBG-2.0 + LBM Relighting (Guided Feed-Forward Pipeline)	Semi-automated pipeline that performs geometry reconstruction, background removal, and latent-space lighting transformation, but allows user adjustments at key control points (e.g., light direction, camera tilt).	Balances technical automation with creative input; realistic blending under controlled relighting; efficient on-device execution; compatible with low compute budgets.	Relies on user interaction for best results; requires strong UX for real-time preview feedback.
C. Lightweight 2D Composite with ML-Aided Filters	Uses traditional 2D filters for contrast, exposure, and soft masking without explicit 3D reasoning.	Extremely fast; minimal processing cost.	Produces flat, unnatural composites; unsuitable for perspective and complex relighting scenarios.

RATIONALE

This option balances automation and user control, enabling creators to guide lighting and perspective adjustments interactively. It produces realistic, high-quality composites without fully replacing human judgment. It leverages advanced AI models while ensuring UX remains responsive. This approach aligns with our vision of AI as a creativity enabler, not a substitute.

IMPACT

Improves user engagement by empowering creative control, thus increasing retention among target users. Differentiates the product with advanced 3D compositing features on mobile platforms. Unlocks monetization potential through premium AI-powered editing tools.Requires optimization of multiple AI models to run efficiently on low-power mobile hardware. Necessitates a responsive user interface for real-time control and previews. Introduces challenges to maintain a balance between inference speed and energy consumption.

DECISION TITLE: Hybrid CLIP + Moondream2 Architecture for Context Aware Suggestions

OWNER ML Lead **DATE:** 21/11/2025 **APPROVED**

PROBLEM

Pipeline 2 aims to deliver a “hands-free” editing experience that identifies and corrects common image issues–lighting, framing, and balance–without requiring users to diagnose defects themselves. Existing single-model defect detection methods (either retrieval-based or generative models) exhibited limitations in accuracy, interpretability, and latency on mobile hardware. A hybrid system design was needed to achieve reliable, context-aware defect detection that could operate efficiently on-device or in lightweight cloud inference modes.

OPTION	DESCRIPTION	PROS	CONS
A. CLIP-only Retrieval-Based Defect Detection	Uses CLIP embeddings to compare the image against a curated list of defect descriptions and rank the top likely issues based on cosine similarity.	Fast inference, low compute cost, easy to fine-tune, interpretable.	Lacks context validation; prone to false positives on complex compositions or stylized imagery.
B. Moondream2-Only Vision-Language Analysis	Employs Moondream2 directly to analyze the image and describe photographic defects in natural language.	Strong visual reasoning, human-readable explanations, self-contained model.	High latency on low-compute devices, limited consistency across diverse defect categories, and larger memory footprint.
C. Hybrid CLIP + Moondream2 (Chosen Option)	Combines CLIP for lightweight retrieval of top defect candidates, followed by Moondream2 for verification and explanation of those candidates.	Balances speed and accuracy, reduces false positives, provides context-aware validation, interpretable outputs, scalable for mobile inference.	Slightly increased integration complexity and setup time for embedding vocabulary management.

RATIONALE

The hybrid model architecture offered the best balance between accuracy, interpretability, and efficiency, critical for the mobile-first and energy-conscious goals of the 2030 AI editor. CLIP efficiently narrows down candidate defects using vector retrieval, minimizing compute overhead, while Moondream2 verifies and contextualizes these results with rich visual reasoning. This combination ensures fast, reliable feedback and grounded textual explanations that improve user trust and usability. The approach aligns with the long-term vision of assistive AI that enhances, not replaces, creative decision-making and supports on-device AI workflows that are both sustainable and scalable.

IMPACT

The decision to use Hybrid CLIP + Moondream2 is expected to be net-positive across user experience, technical feasibility, and business outcomes, because it delivers fast, targeted, and explainable defect analysis that feels intelligent rather than “black box” to creators. It keeps inference lightweight and composable for low-compute devices, while still achieving high-quality, interpretable results that can drive one-tap suggestions, framing fixes, and relighting presets in a unified pipeline. For creators, this translates into fewer failed edits, clearer guidance, and higher confidence in automation; for engineering, it provides a scalable, extensible architecture that can absorb new defect types and reuse components across features; and for the business, it establishes a differentiated, future-ready editing experience that can improve engagement, retention, and the perceived premium value of the product.

DECISION TITLE: Adoption of MagicQuill Unified Stroke-Driven Editing Pipeline

OWNER ML Lead **DATE:** 30/11/2025 **APPROVED**

PROBLEM

To achieve the 2030 vision of a lightweight, mobile-first AI image editor, the team needed to solve the challenge of making image editing as natural and semantic as sketching on a photo. Traditional editing paradigms required complex mask selections, prompt tuning, or manual fine-tuning, creating friction for casual and mobile users. The opportunity was to design a stroke-based pipeline that interprets quick, rough user gestures (e.g., circle, erase, recolor strokes) into meaningful image edits, enabling gesture-level creativity rather than parameter-driven workflows.
This decision focused on selecting the optimal technical pipeline for enabling human-intent recognition and semantic image transformation using AI within strict performance, accuracy, and compute constraints.

OPTION	DESCRIPTION	PROS	CONS
A. Mask-based manual editing with inpainting tools (baseline)	Users manually select regions using SAM and apply localized edits through existing diffusion-based inpainting.	Reliable control; well-understood implementation; compatible with existing workflows.	Time-intensive; not intuitive for non-expert users; lacks semantic understanding of gestures; poor fit for mobile, fast-iteration use.
B. Prompt-only editing using diffusion transformers	Users describe edits entirely through text prompts; model modifies image accordingly.	Intuitive via natural language; low interaction friction.	Low spatial precision; difficult to control regionally; ambiguous interpretation of creative intent.
C. MagicQuill stroke-driven pipeline (LLaVA-1.5 + fine-tuned Stable Diffusion 1.5)	Combines user strokes and image context with a multimodal LLaVA “Draw & Guess” intent interpretation model, followed by localized diffusion-based regeneration.	Natural, gesture-first interface; interprets abstract user actions semantically; high spatial control; supports low-latency inference; enhances creative fluidity.	Requires specialized fine-tuning for accuracy in intent recognition; new UX learning curve for users; dependency on hybrid ML model stack.

RATIONALE

Option C best aligns with the project’s design-first and mobility-focused vision for 2030. The MagicQuill pipeline interprets low effort user input into complex, semantically rich edits like aging faces, altering objects, or changing materials with minimal text input or manual control. By combining LLaVA’s multimodal intent recognition with a fine-tuned Stable Diffusion 1.5 model, it bridges the gap between human gestures and model understanding, enabling real-time, intuitive editing through natural strokes. Its targeted mask generation and optimized diffusion backbone deliver high performance on-device with low power consumption, while seamless integration with modules like SAM, Inpaint4Drag, and Generative Expand ensures unified multimodal interoperability. The compact, modular design supports scalable personalization and quantized inference, aligning with the vision of fast, natural, and energy-efficient creative editing for 2030.

IMPACT

The MagicQuill pipeline delivers a fast editing experience compared to traditional mask-based workflows, dramatically reducing creative friction by allowing users to sketch rather than script edits. Its modular multimodal architecture combining LLaVA intent inference with diffusion control branches, supports efficient on-device execution using quantized, low-VRAM models. This design extends the editor’s capabilities into advanced semantic use cases such as aging, texture transformation, and detailed retouching, which were previously unattainable with regional only methods. By streamlining inference and minimizing manual correction, the system reduces computational cost and user effort, embodying a seamless balance of speed, scalability, and creative freedom.

DECISION TITLE: Adoption of Personalized FLUX.1 Dev Pipeline for Identity-Aware Image Generation

OWNER ML Lead **DATE:** 1/12/2025 **APPROVED**

PROBLEM

To deliver personalized, high-fidelity image editing on mobile and low-compute devices, the product team needed a generative pipeline that balances speed, memory efficiency, and identity accuracy. Traditional diffusion models (full-precision or even 8-bit quantized) were too compute heavy for edge deployment, causing slow inference, high GPU demand, and degraded UX on consumer hardware.
The goal was to design a solution allowing real-time personalization that enables users to generate identity-consistent images with minimal training data and limited computational resources.

OPTION	DESCRIPTION	PROS	CONS
A. Full-precision Diffusion (Baseline)	Use standard full-precision UNet-based diffusion for personalization	High image fidelity and model stability; widely supported ecosystem	Extremely VRAM-intensive (~12–16GB+); unsuitable for mobile; slow inference (>20s/image)
B. LoRA-only Fine-Tuning (8-bit)	Employ 8-bit quantized diffusion with LoRA fine-tuning on UNet	Moderate compute efficiency; faster personalization; retains some fidelity	Still memory-heavy for mid-range devices; identity accuracy degrades at higher compression
C. FLUX.1-Dev + Nunchaku SVD-Quantized 4-bit + LoRA Merge	Integrate Nunchaku's 4-bit quantized diffusion transformer with personalized LoRA fine-tuning and merging pipeline	2× faster inference; low memory use; maintains photorealism and identity; scalable for consumer GPUs; optimized for edge	Slightly increased model complexity during merge; limited open-source tool support for debugging

RATIONALE

Option C best meets both user and system-level requirements for a 2030-class mobile AI editor. The quantized 4-bit diffusion backbone drastically reduces GPU and battery consumption while maintaining photorealistic detail. The personalized LoRA fine-tuning allows the system to rapidly learn user identity from 12 photos without overfitting, ensuring consistent likeness across generated or edited images.
Merging the LoRA weights into Nunchaku's transformer at inference time eliminates the need for runtime fine-tuning, resulting in a lightweight, portable model deployable on consumer GPUs and potentially on-prem mobile AI chips. This approach aligns with the product's vision for fast, intuitive, and energy-efficient creative editing by 2030.

IMPACT

Performance: 2× faster generation times; VRAM reduction.
Scalability: Enables personalized models to run efficiently on mid-tier GPUs .
UX: Seamless identity-aware editing; near-instant personalized outputs (<5s per render).
System Architecture: Simplifies inference path through model merging; reduces runtime dependencies.
Cost: Lower cloud GPU costs by up to 50% for heavy personalization workflows.
Future Alignment: Supports modular personalization and device-level deployment roadmap for 2030.

DECISION TITLE: Rejection of Integrated Portrait Editing Features

OWNER ML Lead **DATE:** 2/12/2025 **REJECTED**

PROBLEM

As part of the 2030 lightweight AI image editor roadmap, the team explored adding a "portrait-specific edits" feature—targeting advanced facial refinements, skin tone matching, expression manipulation, and depth-based relighting. The concept aimed to enhance personalization but introduced overlapping functionality with existing regional editing, inpainting, and style-transfer tools. The addition risked technical bloat and disproportionate compute demands for a narrow user segment, conflicting with the product's lightweight, energy-efficient design goals.

Option	Description	Pros	Cons
A. Dedicated Portrait Pipeline	Build a separate AI pipeline for portrait enhancement using specialized models (e.g., facial landmark refinement, portrait LoRAs, relighting models).	Highly detailed control for portrait photographers and beauty editors. Strong marketing appeal for portrait niche.	Adds significant inference overhead (~40–60% increased compute usage). High complexity in model orchestration. Overlaps heavily with existing edit modules (SAM + Inpaint4Drag). Niche impact versus development effort.
B. Integrated Portrait Features within Existing Editing Stack	Extend current workflows (SAM + Lightning Drag + LeDits++) to handle facial segments more intelligently rather than adding a separate pipeline.	Efficient reuse of infrastructure. Scalable and flexible implementation.	Only partial parity with full portrait enhancement tools. Moderate additional UX complexity.
C. Defer Portrait-Specific Enhancements to a Future Release	Focus current cycle on general-purpose editing and performance; revisit specialized portraits once device compute and power budgets increase.	Keeps MVP lightweight and aligned with mobile-first constraints. Preserves roadmap flexibility.	Delays advanced features for a subset of potential users.

RATIONALE

The feature was rejected for the current release due to low impact relative to engineering and compute costs. The dedicated portrait pipeline conflicted with the app's core principle of lightweight, low-energy editing suitable for 2030 mobile devices. Existing modular tools (SAM, Inpaint4Drag, Lightning Drag) already enable localized, high-control adjustments applicable to faces without bespoke model overhead. Additionally, user research showed limited demand outside professional creator circles. Deferring this feature ensures focus on reliability, general edit speed, and inference efficiency—key differentiators for the MVP stage.

IMPACT

Minimal negative impact on general user experience. Rejecting the feature streamlines the tech stack, reduces model dependencies, and keeps the application efficient for low-compute environments. Future inclusion (post-launch) may be considered once the base editor is optimized and inference pipelines support dynamic model loading without energy cost trade-offs.