RTX 3060 Image Manipulation - Implementation Plan

System: Windows 11, RTX 3060 12GB, 32GB RAM
Current Performance: 35s init / 17s generation (SDXL Base)
Storage: SATA HDD (2TB NVMe incoming)

YOUR GOALS

1. ✓ **Background replacement** - Replace backgrounds seamlessly
2. ✓ **Person/object removal** - Clean removal from images
3. ✓ **Old photo restoration** - Repair and colorize damaged photos
4. ✓ **Colorization** - Add color to B&W photos
5. ✓ **Inpainting/outpainting** - Fill missing areas, extend images
6. ✓ **4K upscaling** - Final output at high resolution
7. ✓ **Style transfer** - Turn people into puppets, cartoons, etc.

PHASE 1: Foundation (COMPLETE ✓)

✓ Step 1.1: Install SDXL Base Model

Status: DONE
Result: 35s init / 17s generation at 1024x1024
Files:

- `sd_xl_base_1.0.safetensors` (6.94 GB) in `models/checkpoints/`

Performance Baseline:

- VRAM usage: ~8.5 GB (no offloading)
- Quality: Excellent
- Speed: 2x faster than previous setup

PHASE 2: Speed Optimization (TODAY)

Goal: Reduce generation time from 17s to 5-8s for quick iterations

Step 2.1: Download SDXL Turbo

Why: 1-step generation = 3-4 seconds per image
Use case: Rapid prototyping, testing ideas, batch processing

Download:

- URL: https://huggingface.co/stabilityai/sdxl-turbo/resolve/main/sd_xl_turbo_1.0_fp16.safetensors
- Size: 6.94 GB
- Place in: `models/checkpoints/`

Workflow Settings (Different from Base):

- Steps: **1** (not 25)
- CFG: **0.0** (not 7.0)
- Sampler: euler_a
- Scheduler: simple

Expected Result: 3-4 seconds per image

Step 2.2: Test SDXL Turbo Workflow

JSON Workflow:

```
json

{
  "checkpoint": "sd_xl_turbo_1.0_fp16.safetensors",
  "steps": 1,
  "cfg": 0.0,
  "sampler": "euler_a",
  "scheduler": "simple"
}
```

Test Prompts:

1. "a red sports car in a parking lot"
2. "portrait of a woman, professional photo"
3. "mountain landscape, golden hour"

Measure: Generation time should be 3-5 seconds

Step 2.3: Decision Point - Base vs Turbo

Use SDXL Base when:

- Final quality outputs
- Complex scenes
- Photorealistic requirements

Use SDXL Turbo when:

- Testing ideas quickly
- Batch processing (50+ images)
- "Good enough" quality acceptable

Strategy: Use Turbo for iteration, Base for finals

PHASE 3: Inpainting Setup (THIS WEEK)

Goal: Remove people/objects and fill backgrounds naturally

Step 3.1: Download SDXL Inpainting Model

Download:

- URL: https://huggingface.co/diffusers/stable-diffusion-xl-1.0-inpainting-0.1/resolve/main/sd_xl_inpainting_0.1.safetensors
- Size: 6.94 GB
- Place in: `models/checkpoints/`

This model is specifically trained for:

- Removing objects
- Removing people
- Filling backgrounds
- Seamless inpainting

Step 3.2: Build Basic Inpainting Workflow

Workflow Structure:

```
Load Image → Draw Mask → Inpainting Model → VAE Decode → Save
```

Key Nodes:

1. **Load Image** - Your source photo
2. **Load Image (Mask)** - Black/white mask (white = area to replace)
3. **VAE Encode** - Convert to latent
4. **CheckpointLoader** - Load inpainting model
5. **KSampler** - Denoise: 0.9-1.0 for full replacement
6. **VAE Decode** - Back to pixels
7. **Save Image**

Test Case:

- Load photo with person
- Mask person in white
- Prompt: "empty street, daytime, photorealistic"
- Result: Person removed, background filled naturally

Expected Time: 12-18 seconds per inpaint

Step 3.3: Test Person Removal

3 Test Images:

1. Person in front of building → Remove person
2. Object on table → Remove object
3. Car in driveway → Remove car

Success Criteria:

- No visible seams
- Natural lighting match
- Background coherent

PHASE 4: ControlNet Integration (THIS WEEK)

Goal: Precise control over composition and structure

Step 4.1: Download ControlNet Models

Required ControlNets:

Depth ControlNet:

- URL: https://huggingface.co/diffusers/controlnet-depth-sdxl-1.0/resolve/main/diffusion_pytorch_model.safetensors
- Rename to: `sdxl_controlnet_depth.safetensors`
- Place in: `models/controlnet/`
- Use: Preserve 3D structure, depth-aware replacements

Canny ControlNet:

- URL: https://huggingface.co/diffusers/controlnet-canny-sdxl-1.0/resolve/main/diffusion_pytorch_model.safetensors
- Rename to: `sdxl_controlnet_canny.safetensors`
- Place in: `models/controlnet/`
- Use: Preserve edges, line art, sharp boundaries

Step 4.2: Build Background Replacement Workflow

Workflow: Subject Preservation + Background Swap

Structure:

```
Load Image → Depth Preprocessor → ControlNet Depth → Text Prompt → Generate
```

Example:

- Input: Person on city street
- Mask: Everything except person
- Depth ControlNet: Preserves person's shape/position
- Prompt: "tropical beach, sunset, palm trees"
- Output: Same person, new background

Expected Time: 15-20 seconds

Step 4.3: Test Background Replacement

Test Cases:

1. Person indoors → Move to outdoor scene
2. Product on white background → Place in lifestyle setting
3. Portrait with busy background → Clean studio background

Success Criteria:

- Subject unchanged
- Lighting matches reasonably
- Perspective correct
- No artifacts at edges

PHASE 5: Photo Restoration Pipeline (NEXT WEEK)

Goal: Restore and colorize old/damaged photos

Step 5.1: Download Face Restoration Models

CodeFormer (Best for faces):

- URL: https://github.com/sczhou/CodeFormer/releases/download/v0.1.0/codeformer.pth
- Place in: `models/facerestore_models/`
- Use: Enhance face details, fix blur

GFPGAN (Alternative):

- URL: https://github.com/TencentARC/GFPGAN/releases/download/v1.3.4/GFPGANv1.4.pth
- Place in: `models/facerestore_models/`
- Use: Face enhancement, good for old photos

Step 5.2: Build Restoration Workflow

Multi-Stage Pipeline:

Stage 1: Denoise & Repair

- Load damaged image
- Inpainting model fills tears/scratches
- Mask damages in white

Stage 2: Colorization

- SDXL with prompt: "restored vintage photograph, natural colors"
- Image-to-image with low denoise (0.3-0.5)

Stage 3: Face Enhancement

- FaceDetailer node (Impact Pack)
- CodeFormer on detected faces
- Upscale faces 2x

Stage 4: Final Upscale

- Ultimate SD Upscale
- 2x or 4x final resolution

Expected Time: 45-90 seconds per photo (multi-stage)

Step 5.3: Test Photo Restoration

Test Images:

1. B&W photo with scratches
2. Faded color photo
3. Damaged portrait with tears

Success Metrics:

- Scratches removed
- Natural colorization
- Faces clear and detailed
- Grain preserved (looks authentic)

PHASE 6: Upscaling Pipeline (NEXT WEEK)

Goal: Output final images at 4K resolution

Step 6.1: Download Upscaling Models

Real-ESRGAN (Best General):

- URL: https://github.com/xinntao/Real-ESRGAN/releases/download/v0.2.5.0/realesr-general-x4v3.pth
- Place in: `models/upscale_models/`
- Use: 4x upscale, photorealistic

ESRGAN 4x (Alternative):

- URL: https://github.com/xinntao/ESRGAN/releases/download/v0.2.1/ESRGAN_SRx4_DF2KOST_official-ff704c30.pth
- Place in: `models/upscale_models/`
- Use: Sharp details, good for text

Step 6.2: Integrate Upscaling into Workflow

Method 1: Simple Upscale (Fast)

```
Generate 1024x1024 → Upscale Model → 4096x4096 → Save
```

Method 2: Ultimate SD Upscale (Better Quality)

```
Generate 1024x1024 → Tile Split → Upscale Each Tile →
SDXL Refine (low denoise) → Merge Tiles → 4096x4096
```

Expected Time:

- Simple: 10-15 seconds
- Ultimate: 60-90 seconds

Step 6.3: Test Upscaling Pipeline

Test Cases:

1. Portrait $1024^2 \rightarrow 4096^2$ (check face details)
2. Landscape $1024^2 \rightarrow 4096^2$ (check textures)
3. Product photo $1024^2 \rightarrow 4096^2$ (check edges)

Quality Check:

- No pixelation
- Sharp details
- No AI artifacts
- Natural grain

PHASE 7: Style Transfer (FUN STUFF!)

Goal: Transform people into puppets, cartoons, paintings

Step 7.1: Download Style LoRAs

LoRA (Low-Rank Adaptation): Small model files (50-200 MB) that add styles

Recommended LoRAs:

- **Puppet/Muppet style:** Search CivitAI "puppet style SDXL"
- **Pixar/3D style:** Search "pixar cartoon SDXL"
- **Oil painting:** Search "oil painting SDXL"
- **Anime style:** Search "anime style SDXL"

Where to find: https://civitai.com/models?baseModel=SDXL%201.0

Place in: `models/loras/`

Step 7.2: Build Style Transfer Workflow

Using IPAdapter (Best Method):

Workflow:

```
Load Image (person) → IPAdapter (style reference) →
SDXL Base + LoRA → Generate → Save
```

Or Simple Method:

```
Load Image → Image-to-Image →
Prompt: "person as a muppet puppet, studio lighting" →
Denoise: 0.6-0.8 → Generate
```

Expected Time: 15-20 seconds

Step 7.3: Test Style Transfers

Test Subjects:

1. Portrait photo → Muppet puppet
2. Portrait photo → Pixar character
3. Landscape → Oil painting
4. Person → Anime character

Quality Check:

- Recognizable as original subject
- Style applied consistently
- No uncanny valley effects

PHASE 8: Batch Processing & Automation (LATER)

Goal: Process multiple images efficiently

Step 8.1: Set Up Batch Workflow

Batch Image Loader:

- Load folder of images
- Process each through same workflow
- Save with sequential numbering

Use Cases:

- 50 product photos → background removed
- Album of old photos → all restored
- Event photos → all upscaled to 4K

Step 8.2: Script Integration (Optional)

Python Script to:

1. Watch folder for new images
2. Auto-load into ComfyUI workflow
3. Process and save to output folder
4. Send notification when complete

PHASE 9: NVMe Migration (WHEN IT ARRIVES)

Goal: Reduce cold start from 35s to 10-15s

Step 9.1: NVMe Setup

Installation:

1. Install NVMe in M.2 slot
2. Initialize in Disk Management
3. Format as NTFS
4. Assign drive letter (E:)

Step 9.2: Move ComfyUI

Migration Steps:

1. Close ComfyUI completely
2. Copy entire folder: `D:\Misc\ComfyUI` → `E:\ComfyUI`
3. Update shortcuts/bat files
4. Test launch from new location
5. Delete old folder after confirming it works

Expected Improvement:

- Cold start: 35s → 10-15s
- Model switching: Faster
- No more 16-minute first loads

Step 9.3: Move Ollama Models Too

Set environment variable:

```
OLLAMA_MODELS=E:\Ollama_Models
```

Both ComfyUI and Ollama on NVMe = Fast everything

WEEKLY SCHEDULE

Week 1 (This Week)

- [x] **Day 1:** SDXL Base working (DONE - 35s/17s)
- [ ] **Day 2:** Download SDXL Turbo, test speed (target: 3-5s)
- [ ] **Day 3:** Download inpainting model
- [ ] **Day 4:** Build inpainting workflow, test person removal
- [ ] **Day 5:** Download ControlNet Depth
- [ ] **Day 6:** Build background replacement workflow
- [ ] **Day 7:** Test & refine workflows

Week 2

- [ ] **Day 8:** Download face restoration models
- [ ] **Day 9:** Build photo restoration workflow
- [ ] **Day 10:** Test on 5 old photos
- [ ] **Day 11:** Download upscaling models
- [ ] **Day 12:** Integrate upscaling into workflows
- [ ] **Day 13:** Download style LoRAs
- [ ] **Day 14:** Test style transfers (puppets, cartoons)

Week 3

- [ ] **Day 15-17:** Refine all workflows based on testing
- [ ] **Day 18-19:** Build batch processing setup
- [ ] **Day 20-21:** Document personal workflow library

Week 4 (When NVMe Arrives)

- [ ] **Day 22:** Install NVMe
- [ ] **Day 23:** Migrate ComfyUI and Ollama
- [ ] **Day 24:** Benchmark improvements
- [ ] **Day 25-28:** Explore advanced techniques

SUCCESS METRICS

Performance Targets

- [x] **Base generation:** 15-20 seconds ✓ (Got 17s)
- [ ] **Turbo generation:** 3-5 seconds
- [ ] **Inpainting:** 12-18 seconds
- [ ] **Background replace:** 15-20 seconds
- [ ] **Photo restoration:** 45-90 seconds (multi-stage)
- [ ] **4K upscale:** 10-15 seconds (simple) / 60-90s (ultimate)
- [ ] **Cold start (NVMe):** 10-15 seconds

Quality Targets

- [ ] **Inpainting:** No visible seams, natural fill
- [ ] **Background replace:** Lighting matches, no artifacts
- [ ] **Restoration:** Natural colors, clear faces
- [ ] **Upscaling:** Sharp at 4K, no pixelation
- [ ] **Style transfer:** Recognizable, consistent style

DOWNLOADS SUMMARY

Models to Download (In Order)

Phase 2 (Today):

- [ ] SDXL Turbo (6.94 GB)

Phase 3 (This Week):

- [ ] SDXL Inpainting (6.94 GB)

Phase 4 (This Week):

- [ ] ControlNet Depth SDXL (2.5 GB)
- [ ] ControlNet Canny SDXL (2.5 GB)

Phase 5 (Next Week):

- [ ] CodeFormer (376 MB)
- [ ] GFPGAN (348 MB)

Phase 6 (Next Week):

- [ ] Real-ESRGAN 4x (64 MB)

Phase 7 (Later):

- [ ] 3-5 Style LoRAs (50-200 MB each)

Total Additional Downloads: ~22 GB

Current Disk Space:

- SDXL Base: 6.94 GB
- Total needed: ~29 GB
- Have space? (Check before starting)

TROUBLESHOOTING CHECKPOINTS

If Generation Time Increases

Check:

- Is another app using GPU? (Ollama, Chrome hardware acceleration)
- Close unnecessary apps
- Check Task Manager → GPU usage should be 95-100% during generation

If Quality Drops

Check:

- Using correct model? (Base for quality, Turbo for speed)
- CFG too low? (Should be 7.0 for Base, 0.0 for Turbo)
- Steps too low? (25 for Base, 1 for Turbo)

If VRAM Issues Return

Check:

- Did you accidentally load the wrong model?
- Console should show NO "offloaded" messages
- SDXL should use ~8.5 GB max

If Workflow Fails

Check:

- All nodes connected?
- Missing custom nodes? (Install via Manager)
- Model in correct folder?
- Restart ComfyUI and try again

BACKUP STRATEGY

Save Your Working Setups

After Each Phase:

1. Export working workflows as JSON
2. Save to: `E:\ComfyUI_Workflows\Backups\`
3. Name clearly: `SDXL_Inpainting_PersonRemoval_v1.json`

Keep Notes:

- What settings worked best
- Which prompts gave best results
- Any special tricks discovered

RESOURCES

Learning Materials

- **ComfyUI Workflows:** https://openart.ai/workflows
- **SDXL Tutorials:** https://www.youtube.com/results?search_query=comfyui+sdxl+tutorial
- **CivitAI Models:** https://civitai.com/models?baseModel=SDXL%201.0
- **Reddit Community:** r/comfyui

Model Repositories

- **HuggingFace:** https://huggingface.co/models?pipeline_tag=text-to-image&sort=trending
- **CivitAI:** https://civitai.com
- **ComfyUI Examples:** https://comfyanonymous.github.io/ComfyUI_examples/

FINAL NOTES

You Are Here: Phase 1 Complete ✓
Next Action: Download SDXL Turbo (Phase 2.1)
Timeline: 4 weeks to full capability
Current Performance: 35s init / 17s gen (Excellent baseline!)

Remember:

- Test each phase before moving to next
- Document what works for you
- Take breaks - don't rush
- Enjoy the process!

This is now a realistic, achievable plan with your hardware.

Start Phase 2 whenever you're ready!