

Enhancement Timeout Issue - Request for Claude Code Direction

Current Status

After implementing Claude Code's last set of changes (gpt-image-1.5 model, new prompts, UI updates), **everything was working great and photos looked much better**.

However, after a subsequent change to fix the aspect ratio (from 4:3 to 3:2), we're now experiencing **504 timeout errors**.

IMPORTANT: The timeout started AFTER the aspect ratio change, not after the model/prompt changes.

What Was Implemented (From Claude Code's Last Instructions)

1. Model Change

- Switched from `flux-kontext` to `gpt-image-1.5`
- Reasoning: Better for photo editing vs. generation

2. New Prompts

- "Edit this existing photo" framing (not "enhance" or "generate")
- Lightroom/Photoshop editing style instructions
- Room-specific texture hints
- Strong "DO NOT add/remove objects" restrictions

3. Creative Additions as Opt-in Checkboxes

- `addTowels` , `smoothLinens` , `addToiletPaper` are now checkboxes
- NOT included in base prompts

4. "Edit Prompt" Button

- Shows the ACTUAL prompt being sent to the API
- Admin can edit before running enhancement

5. Aspect Ratio Fix

- Changed from 4:3 to 3:2 (Airbnb optimal per spec)

6. Stale URL Cleanup Route

- Created `/api/admin/cleanup-stale-urls` to clear old external URLs
 - User ran this successfully (cleared 1 photo)
-

Current Error

User's Screenshot Shows:

listinglift.abacusai.app says
Enhancement timed out. The image may be too complex. Try again or use a lower intensity.

Console Errors:

Failed to **load** resource: the server responded with a status of 504 () enhance:1

Access to image at 'https://abacusai-apps-92d1307...' from origin 'https://listinglift.abacusai.app' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header **is** present on the requested resource.

Failed to **load** resource: net::ERR_FAILED (multiple S3 URLs)

Current Enhancement Route Code

File: /app/api/photos/[id]/enhance/route.ts

```
export const maxDuration = 120; // Allow up to 2 minutes

// Uses AbortController for 90-second client-side timeout
const controller = new AbortController();
const timeoutId = setTimeout(() => controller.abort(), 90000);

const response = await fetch("https://apps.abacus.ai/v1/chat/completions", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
    Authorization: `Bearer ${process.env.ABACUSAI_API_KEY}`,
  },
  body: JSON.stringify({
    model: "gpt-image-1.5",
    messages: [
      {
        role: "user",
        content: [
          { type: "image_url", image_url: { url: originalUrl } },
          { type: "text", text: prompt },
        ],
      },
    ],
    modalities: ["image"],
    image_config: {
      num_images: 1,
      aspect_ratio: photo.orientation === "portrait" ? "2:3" : "3:2",
    },
    max_tokens: 1000,
  }),
  signal: controller.signal,
});
```

Analysis of the Problem

1. Timeout Issue

- `gpt-image-1.5` is taking longer than 90 seconds to process
- The 504 Gateway Timeout suggests the request is exceeding server limits
- Previously, `flux-kontext` completed in ~4-12 seconds

2. CORS Errors

These appear AFTER the timeout error, suggesting:

- Old enhanced URLs in the database are still pointing to external S3
- OR the cleanup didn't catch all of them
- OR the browser is caching old URLs

3. Model Speed Comparison

Model	Typical Speed	Quality
flux-kontext	4-12 seconds	More generative, added fake objects
gpt-image-1.5	60-120+ seconds?	Unknown - timing out before completion

Questions for Claude Code

1. Is `gpt-image-1.5` the right model for this use case?

- It's timing out at 90 seconds
- Is there a faster alternative that still maintains photorealism?
- Should we try a different model like `gpt-image-1` (without the .5)?

2. What's the actual expected processing time for `gpt-image-1.5` ?

- Is 90 seconds too short of a timeout?
- Should we increase to 180 seconds or implement a polling/webhook approach?

3. Alternative approaches:

- Can we use a streaming response to keep the connection alive?
- Should we implement a background job pattern with status polling?
- Is there a "quality" or "speed" parameter we can adjust?

4. Fallback strategy:

- If `gpt-image-1.5` continues timing out, what's the best fallback?
- Can we tune `flux-kontext` with stronger restrictions instead?

5. CORS errors:

- How should we handle the persistent CORS errors after hard refresh?
 - Should we modify how we display/load enhanced images?
-

Environment Details

- **Platform:** Abacus AI DeepAgent
 - **API:** Abacus AI RouteLLM API (<https://apps.abacus.ai/v1/chat/completions>)
 - **Deployment:** Cloudflare-fronted (504 from Cloudflare)
 - **Current Timeout:** 90 seconds (AbortController) + 120 seconds (Next.js maxDuration)
-

What We Need From Claude Code

1. **Model recommendation** - What's the best model for real estate photo enhancement that:
 - Completes in under 60 seconds
 - Maintains photorealism
 - Doesn't add fake objects
 - Works with the Abacus AI API
 2. **Code changes** - Specific changes to fix:
 - The timeout issue
 - The CORS errors (if model-related)
 - Any prompt adjustments needed for the recommended model
 3. **Architecture recommendation** - If longer processing times are unavoidable:
 - Should we implement async processing with status polling?
 - How should we handle this in the UI?
-

Files That May Need Changes

1. `/app/api/photos/[id]/enhance/route.ts` - Main enhancement logic
 2. `/app/api/submissions/[id]/auto-enhance/route.ts` - Background auto-enhancement
 3. `/lib/enhancement-prompts.ts` - If prompts need adjustment for different model
 4. `/app/admin/submissions/[id]/page.tsx` - If UI needs to handle async status
-

Previous Enhancement History

Attempt	Model	Aspect Ratio	Result
1	Manual prompts	-	Never completed
2	gpt-image-1	-	524 Cloudflare timeout + CORS
3	flux-kontext	4:3	✅ Fast (~4-12s) but added fake objects
4	gpt-image-1.5	4:3	✅ WORKED GREAT - Photos looked much better
5	gpt-image-1.5	3:2	❌ 504 timeout (current)

Key Insight: The ONLY change between attempt 4 (working) and attempt 5 (broken) was the aspect ratio change from 4:3 to 3:2 .

The Aspect Ratio Change (Suspected Cause)

Before (Working):

```
aspect_ratio: photo.orientation === "portrait" ? "3:4" : "4:3",
```

After (Broken):

```
aspect_ratio: photo.orientation === "portrait" ? "2:3" : "3:2",
```

Why we made this change: The spec document says Airbnb optimal photo dimensions are 3000×2000 px (3:2 ratio). But this change appears to have broken the enhancement.

Question for Claude Code: Is 3:2 not a supported aspect ratio for gpt-image-1.5 ? Should we revert to 4:3 or use a different ratio?

Proposed Quick Fix (For Testing)

If Claude Code agrees, we can quickly revert the aspect ratio to confirm this is the cause:

```
// Revert to working aspect ratio
aspect_ratio: photo.orientation === "portrait" ? "3:4" : "4:3",
```

If this fixes the timeout, we'll know the 3:2 aspect ratio is the issue, and we can then discuss alternatives for getting the Airbnb-optimal dimensions.

Please provide direction on the best path forward.