Image Generation / Midjourney

Identify successful remix patterns from existing images and generate variant parameters — turns one good image into a repeatable system.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Series Creation, Variation Systems
Updated: May 2026
Why This Prompt Exists
You get one great image. Then you can’t replicate it. Or you want 10 variations but don’t know which parameters to change. Remix mode solves this — but only if you know the patterns.

You get:

- one good image, then nothing else (can’t replicate the style)
- variations that lose what made the original good
- no understanding of which parameters create which changes
- wasted remix attempts changing the wrong things
- inability to build a series with consistent quality

But remix patterns can be detected:

- subject changes: what happens when you change the subject noun
- style variations: what changes with –sref adjustments
- composition shifts: what changes with –ar adjustments
- detail level: what changes with –stylize values
- color palette: what happens with weighted color terms

Without pattern detection, remix is random.

This prompt identifies successful remix patterns and generates variant parameters.

The Prompt

```
Assume the role of a Midjourney remix analyst who identifies variation
patterns.

Your task is to analyze a successful image and generate remix
parameters for variants.

Generate:

1. SOURCE IMAGE ANALYSIS
   - Original prompt (if available): [paste]
   - Key characteristics to preserve: [composition, lighting, style,
mood]
   - Parameters used: [--ar, --s, --c, --sref, etc.]

2. REMIX DIMENSIONS TO EXPLORE

| Dimension | Variation Options | How to Change | Expected Effect |
|-----------|-------------------|---------------|-----------------|
| Subject | [list alternatives] | Change noun in prompt | Different
focal object |
| Composition | [portrait/landscape/close-up/wide] | Add framing terms
| Different visual structure |
| Color palette | [warm/cool/monochromatic/vibrant] | Adjust color
terms or --sref | Different mood |
| Detail level | [more/less detail] | Adjust --stylize | More/less
```

interpretation |

3. REMIX PATTERN MATRIX

Remix Type	Parameter Changes	Seed Strategy	Expected Similarity
Same subject, different style	Change --sref only	Same seed	60-70% similar
Different subject, same style	Change subject noun	Same seed	50-60% similar
Same subject, different composition	Add framing, change --ar	Different seed	40-50% similar
Series exploration	Vary --c from 10-50, keep subject	Fixed seed	30-70% range

4. VARIANT GENERATION PROMPTS

Variant 1: [type]
`[modified prompt] --seed [X] --ar [Y] --s [Z]`

Variant 2: [type]
`[modified prompt] --seed [X] --ar [Y] --s [Z]`

Variant 3: [type]
`[modified prompt] --seed [X] --ar [Y] --s [Z]`

5. SEED STRATEGY

```
   - Preserve seed for style consistency across variants
   - Change seed for composition or subject changes
   - Fixed seed + chaos = controlled exploration

6. SUCCESS METRICS FOR REMIX
   - Preserves: [what must stay the same across the series]
   - Varies: [what can change]
   - Rejection criteria: [what would make a variant unacceptable]

INPUTS:

Successful image description (or upload if possible):
[DESCRIBE THE IMAGE — subject, style, composition, mood]

Original prompt (if known):
[PASTE OR "UNKNOWN"]

Number of variants needed:
[E.G., "10 variants for a social media series"]

What must stay consistent:
[E.G., "The character's face, the lighting style"]

What can vary:
[E.G., "Background, clothing, pose, color palette"]

RULES:
- Always preserve the seed when you want style consistency
- Change the seed when you want completely different composition
```

```
- Use chaos (--c) for exploration within a fixed seed
- Test one variable at a time to understand its effect
- Document successful remix patterns for reuse
- Remix mode must be enabled in settings for --seed changes to work
```

How To Use It

- Always preserve the seed when you want style consistency across variants.
- Change the seed when you want completely different composition or layout.
- Use chaos (–c) for exploration within a fixed seed — controlled variation.
- Test one variable at a time to understand its effect on the output.
- Document successful remix patterns for reuse in future projects.
- Remix mode must be enabled in Midjourney settings for –seed changes to work.

Example Input

Successful image description:
“A warrior in blue armor standing on a cliff at sunset, dramatic lighting, epic fantasy style”

Original prompt:
`blue armor warrior on cliff at sunset –ar 16:9 –s 400 –seed 1234`

Number of variants needed:
“8 variants for a character series”

What must stay consistent:
“The warrior’s face and armor color, the dramatic lighting”

What can vary:
“Background, pose, time of day, weapon, weather”

Why It Works
Most users get one good image and stop — or try to remix by changing everything at once,

losing what made the original work.

This framework improves outcomes by forcing:

- source image analysis (what works, what to preserve)
- remix dimension identification (what can vary)
- pattern matrix creation (which changes produce which effects)
- variant generation (ready-to-use prompts)
- seed strategy (consistency vs. exploration)

Failure modes this prevents:

- One good image, then nothing else (no remix pattern)
- Variations that lose what made the original good (wrong parameters changed)
- Inconsistent series (no preserved seed, no fixed elements)
- Wasted remix attempts (changing too many variables at once)

This improves on: Single-image generation. Remix patterns turn one success into a repeatable system.

Related to: MJ-04 (Chaos) for variation control; MJ-05 (Weighted Prompt) for subject emphasis.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also Chaos Parameter Scaler