

Session Prompts

All user prompts from the Claude Code session for the AIVideo.com assessment.

Prompt 1

Browse startups

Home

›

Companies

›

AIVideo.com

AIVideo.com

The all in one tool for AI powered video production

Summer 2023

Active

Company

Jobs

5

<https://aivideo.com>

The all in one tool for AI powered video production!

Active Founders

Justin Harvey

Justin Harvey

Founder

Building superhuman video production as a service - AIVideo.com

Eduardo Faraday

Eduardo Faraday

Twitter account

Founder

Building AIVideo.com - Best AI for Video Production

Former founder at Workep. Scaled to +300k users. Prev. Product lead at Oasis, first AI Video Avatars. @BerkeleySkydeck | StanfordLB

Orestis Lykouropoulos

Orestis Lykouropoulos

Twitter account

Founder

Greek born, US based software engineer & composer.

Passionate about language applications of ML (ex AWS AI, ex Alexa) and orchestral music.

Jobs at AIVideo.com

View all jobs

Create the AIVideo.com newsletter magazine

Remote (US)

\$2K - \$6K / monthly

0.10% - 0.50%

Any (new grads ok)

Apply Now ›

AIVideo.com seeks Lead Product Designer to reinvent video production

San Francisco, CA, US / Remote

\$80K - \$180K

0.25% - 2.50%

3+ years

Apply Now ›

First Marketing Hire (to make AIVideo.com a household name)

San Francisco, CA, US / Remote (US)

\$3K - \$10K

0.10% - 1.00%

1+ years

Apply Now ›

AIVideo.com seeks pioneering AI Engineer to reinvent video production

San Francisco, CA, US / Remote

\$80K - \$180K

0.25% - 2.50%

3+ years

Apply Now ›

AIVideo.com seeks pioneering Design Engineer to reinvent video production

San Francisco, CA, US / Remote

\$80K - \$180K

0.25% - 2.50%

3+ years

Apply Now ›

AIVideo.com

AIVideo.com

Founded:

2023

Batch:

Summer 2023

Team Size:

7

Status:

Active

Primary Partner:

Harj Taggar

X (Twitter) logo

Footer

Y CombinatorY Combinator

Programs

YC Program

Startup School

Work at a Startup

Co-Founder Matching

Company

[YC Blog](#)

[Contact](#)

[Press](#)

[People](#)

[Careers](#)

[Privacy Policy](#)

[Notice at Collection](#)

[Security](#)

[Terms of Use](#)

[Resources](#)

[Startup Directory](#)

[Startup Library](#)

[Investors](#)

[SAFE](#)

[Hacker News](#)

[Launch YC](#)

[YC Deals](#)

[Make something people want.](#) [Apply](#)

[Twitter](#) Interested in Potential Spring Internship @ AIVideo.com

[Inbox](#)

[AI Overview](#)

Aadarsh inquired about an internship and scheduled an interview with Orestis for Thursday, January 8th, at 10 am PST.

Orestis provided Aadarsh with account credits and requested a standalone project proposal based on product exploration.

Aadarsh suggested three MVP ideas, but Orestis shifted the assignment to a two-hour, recorded, standalone project due to codebase setup complexity.

Aadarsh agreed to the new format, and Orestis confirmed Wednesday at 12 pm PST for the project, sending the spec just before the call.

By Gemini; there may be mistakes. [Learn more](#)

Aadarsh Battula aadarshb123@gmail.com

Tue, Jan 6, 3:38 PM (8 days ago)

to orestis

Hi Orestis,

Hope you're doing well! This is Aadarsh from LinkedIn.

Availabilities:

- Thursday 8am - 6pm PST (1/8)
- Friday 8am - 6pm PST (1/9)
- Saturday 8am - 6pm PST (1/10)
- Sunday 8am - 6pm PST (1/11)

Best,

Aadarsh Battula

Aadarsh Battula aadarshb123@gmail.com

Attachments

Tue, Jan 6, 3:43 PM (8 days ago)

to orestis

One attachment

- Scanned by Gmail

Orestis Lykouropoulos

Tue, Jan 6, 4:33 PM (8 days ago)

to me

Hi Aadarsh,

Let's do Thursday 1/8 10am PST?

Orestis

Aadarsh Battula aadarshb123@gmail.com

Tue, Jan 6, 4:46 PM (8 days ago)

to Orestis

Sounds good! Looking forward to speaking with you!

Best,

Aadarsh Battula

Orestis Lykouropoulos

Thu, Jan 8, 1:25 PM (6 days ago)

to me

Great meeting you! I just gave you a free account with this email - there's a small chance it gets overwritten on signup, I usually use our internal tool after the account exists. Let me know if you don't see the credits after you sign up.

Just to have it writing in case you wanna refer to it:

Play with the product a bit, and come up with a roughly 2 hour MVP feature or improvement, then send me a one paragraph or so description of it. I'll tell you if I think it's a good project for this case and then have you record your screen as you go. You can go longer if you really want to, but I prefer something around 2 hours, give or take.

Best,

Orestis

Aadarsh Battula aadarshb123@gmail.com

Thu, Jan 8, 1:31 PM (6 days ago)

to Orestis

Nice meeting you too! I see the 20k credits on my account. I will send you an email later today with a MVP feature or improvement.

Best,

Aadarsh Battula

Aadarsh Battula aadarshb123@gmail.com

Sat, Jan 10, 2:41 PM (4 days ago)

to Orestis

Hi Orestis,

Here are some ideas:

- Smart Defaults: When a user starts a new project, we can pre-fill settings (style, aspect ratio, etc.) based on their most common choices from previous projects.
- Manual Setting Defaults: When a user starts a new project, we can have a "Clone Settings From Previous Project" button which allows you to select on a previous project to reconfigure the exact settings.
- Prompt Improver: When a user types a short/vague prompt, we can suggest an "enhanced" version that inspires them to include more details.

I think the setting defaults might be a user pain point because users will probably use the same settings to pump out as many pieces as possible. Also, if possible, could I potentially implement one of these features/improvements on a separate branch or fork within the codebase?

Best,

Aadarsh Battula

Orestis Lykouropoulos

Jan 10, 2026, 4:12 PM (4 days ago)

to me

Hi Aadarsh,

I think these are all great ideas and I'm definitely seeing that you have an eye for product as well. I have to apologize though because now that I'm thinking about it more I don't think we're set up to have you work in our codebase quickly. There's a decent amount of set up, mostly on the aws and a couple keys here and there that should be stored better but aren't yet. This one's completely on me, we haven't done many interviews recently and I decided to switch it up without thinking it through fully. I used to do standard algorithm interviews but I think those are becoming obsolete with AI-assisted coding.

So, I still think we should do a project, but standalone. If that sounds good, I'd like to find a time where you have 2 uninterrupted hours, I'll email you the project spec a little bit before and have you record your screen as you go, then send me the recording. I'll be available for clarifying questions right after I send you the spec if you have any. Does that work?

Best,

Orestis

Aadarsh Battula

Jan 10, 2026, 5:19 PM (4 days ago)

Hi Orestis, Sounds good, that works for me! Best,

Orestis Lykouropoulos

Jan 10, 2026, 6:16 PM (4 days ago)

to me

Great let me know what time works next week. I'm pretty open any day other than Tuesday.

Aadarsh Battula aadarshb123@gmail.com

Sat, Jan 10, 6:41 PM (4 days ago)

to Orestis

How does Wednesday at 12 pm pst sound?

Aadarsh Battula aadarshb123@gmail.com

Sun, Jan 11, 5:23 PM (3 days ago)

to Orestis

^^

Orestis Lykouropoulos

Jan 12, 2026, 12:54 PM (2 days ago)

to me

Hi,

Yes let's do Wed 12pm PST, I will send you the project description a few minutes before and we can jump on a call if you have any questions.

Best,

Orestis

Understand this

Prompt 2

understand this:

AIVideo.com

The all in one tool for AI powered video production

AIVideo.com seeks pioneering AI Engineer to reinvent video production

\$80K - \$180K

-

0.25% - 2.50%

-

San Francisco, CA, US / Remote

Job type

Full-time

Role

Engineering, Machine learning

Experience

3+ years

Visa

Will sponsor

Connect directly with founders of the best YC-funded startups.

Apply to role ›

Justin Harvey

Justin Harvey

Founder

About the role

We have collected the best dataset in the world for video editing.

Our popular web-based video editor produces thousands of user actions per minute. The goal is to use that user data to power a human level copilot for video production.

We have already implemented several of the most advanced AI agents for video production, used daily by our 3,500 paid subscribers (\$1.3M/year revenue).

Now we want to hire YOU to take it to the next level:

reinforcement learning w/ our unique dataset

rigorous evals

product gets smarter and more personalized as you use it

In your role you will work directly with the CEO to drive gains across the entire platform.

About AIVideo.com

AIVideo.com is a vertically integrated studio. Every 10 weeks we fly in creators from around the world to make content together in a sweet mansion somewhere. Then we take the feedback and ideas from the most viral creators and work as hard as we can to productize them.

By doing that we have gone from 0 to \$1.3M in annual revenue in just 12 months, with plans to grow 10x this year.

There's no other business like it, and we want talented people like YOU to get involved!

Prompt 3

No, he will send me the project description in a couple minutes, so I will enter it once I get it.

Prompt 4

Orestis Lykouropoulos

2:51 PM (1 minute ago)

to me

Format: 1-2 hours recording your screen as you go. Explain key decisions and thinking process out loud where it makes sense but don't feel like you have to talk the entire time. Aim to finish at the 1 hour mark, but go up to 2 if you need extra time, which can easily happen in an open ended project and I won't hold it against you. Completely open book, meaning you should use AI as much as possible, and can use the internet, google, chatgpt, open source code, etc. Nothing is off limits other than asking other people for help.

Task: Build a standalone one-page image editor with one key feature. I want to be able to hover over different objects/people and show a pixel mask selecting just that. Example:

image.png

(Note: unlike this example I want to show the mask only on hover)

You might have a lot more than 2 objects in one image. It's not required to label the selection as cat/dog/x, just to be able to select it.

I'm looking for something that 1) accomplishes the task, but 2) also is easy to use, so spend some time on UI/UX as well, not just the functionality.

Language/frameworks: completely up to you, the only requirement is that they should be able to scale to production levels (in theory, don't need to load test) and be able to use them commercially. So that means open source code/models are fine if they meet the criteria.

—

I'm around if you have questions now or in the next hour. I would recommend recording with QuickTime or some other software other than Zoom/Google meets cause those can be finicky and take a while to upload the recording after. Also good to do a quick test recording before you start.

Best,

Orestis

Prompt 5

Can you provide more options and outline the key benefits and issues?

Prompt 6

Lets go ahead with option 2. Can you plan out the steps need to implement this? Break this down into byte-sized steps?

Prompt 7

Can you explain what replicate.com is?

Prompt 8

I created an API key for replicate in the .env. It is assigned to REPLICATE_TOKEN. Lets proceed with the basic react app setup first

Prompt 9

I think we need to install tailwind. [plugin:vite:css] [postcss] It looks like you're trying to use `tailwindcss` directly as a PostCSS plugin. The PostCSS plugin has moved to a separate package, so to continue using Tailwind CSS with PostCSS you'll need to install `@tailwindcss/postcss` and update your PostCSS configuration.

```
/Users/aadarshbattula/Projects/aivideo-assessment/src/index.css:undefined:null
```

```
at lt (/Users/aadarshbattula/Projects/aivideo-assessment/node_modules/tailwindcss/dist/lib.js:1:1)
```

```
at LazyResult.runOnRoot (/Users/aadarshbattula/Projects/aivideo-assessment/node_modules/postcss/lib/lazy-result.js:1:1)
```

```
at LazyResult.runAsync (/Users/aadarshbattula/Projects/aivideo-assessment/node_modules/postcss/lib/lazy-result.js:1:1)
```

```
at LazyResult.async (/Users/aadarshbattula/Projects/aivideo-assessment/node_modules/postcss/lib/lazy-result.js:1:1)
```

```
at LazyResult.then (/Users/aadarshbattula/Projects/aivideo-assessment/node_modules/postcss/lib/lazy-result.js:1:1)
```

Click outside, press Esc key, or fix the code to dismiss.

You can also disable this overlay by setting `server.hmr.overlay` to `false` in `vite.config.js`.

Prompt 10

Ok this looks fine for now. Is the react app set up finished?

Prompt 11

Yes proceed with this step.

Prompt 12

This looks good. Can you change it to follow this color palette? AIVIDEO.COM Color Palette

Primary Colors

ColorHex CodeUsageDeep Black#0D0D0D or #121212Main backgroundDark Gray#1A1A1A or #1E1E1ECard/panel backgroundsHot Pink/Magenta#FF2D7C or #FF1F6EPrimary accent (buttons, highlights, “CREATE” text)

Secondary/Supporting Colors

ColorHex CodeUsageWhite#FFFFFFPrimary text, headingsLight Gray#A0A0A0 or #9CA3AFSecondary text, descriptionsMedium Gray#2A2A2A or #333333Card borders, dividersCyan/Teal (banner)#00D1D1 or #00E5E5Top banner link text

Accent Details

ColorHex CodeUsagePink Glow#FF2D7C with opacityGlow effects on “CREATE”Dark Card Hover#252525Card hover states

Prompt 13

Can you center the upload different image button?

Prompt 14

Also, can you break up App.jsx into multiple files for code readability and separation of concerns

Prompt 15

Yes, lets move to that setup. The API key should be accessible by REPLICATE_TOKEN in the root directory.

Prompt 16

b06c281

/tmp/claude/-Users-aadarshbattula-Projects-aivideo-assessment/tasks/b06c281.output

failed

Background command “Start the Express server” failed with exit code 137

Read the output file to retrieve the result: /tmp/claude/-Users-aadarshbattula-Projects-aivideo-assessment/tasks/b06c281.output

Prompt 17

Can you say the command to run the backend server

Prompt 18

Error: Request to <https://api.replicate.com/v1/models/meta/sam-2-base/predictions> failed with status 404 Not Found: {“detail”:“The requested resource could not be found.”,“status”:404} .

lucataco

/

segment-anything-2

Star

Copy

Segment Anything 2 (SAM2) by Meta - Automatic mask generation

Cold

Public

36.8K runs

L40S

License

GitHub

Paper

Playground

API

Examples

README

Versions

Input

Form

JSON

Node.js

Python

HTTP

Cog

Docker

image

image

•

file

<https://replicate.delivery/pbxt/LMUHQoCnNzN15MpwWLPJUVF96g3zzer6p3dNTHtVrNxqMMe0/soccer.png>

Upload a file from your machine

Take a photo with your webcam

Input image

mask_limit

integer

2

maximum number of masks to return. If -1 or None, all masks will be returned.

NOTE: The masks are sorted by predicted_iou.

Default: -1

points_per_side

integer

64

The number of points to be sampled along one side of the image.

Default: 64

points_per_batch

integer

128

Sets the number of points run simultaneously by the model

Default: 128

pred_iou_thresh

number

0.7

A filtering threshold in $[0,1]$, using the model's predicted mask quality.

Default: 0.7

stability_score_thresh

number

0.92

A filtering threshold in $[0,1]$, using the stability of the mask under changes to the cutoff used to binarize the model's mask predictions.

Default: 0.92

stability_score_offset

number

0.7

The amount to shift the cutoff when calculated the stability score.

Default: 0.7

crop_n_layers

integer

1

If >0 , mask prediction will be run again on crops of the image

Default: 1

box_nms_thresh

number

0.7

The box IoU cutoff used by non-maximal suppression to filter duplicate masks.

Default: 0.7

crop_n_points_downscale_factor

integer

2

The number of points-per-side sampled in layer `n` is scaled down by `crop_n_points_downscale_factor**n`.

Default: 2

`min_mask_region_area`

number

25

If `>0`, postprocessing will be applied to remove disconnected regions and holes in masks with area smaller than `min_mask_region_area`.

Default: 25

`mask_2_mask`

boolean

Whether to add a one step refinement using previous mask predictions.

Default: true

`multimask_output`

boolean

Whether to output multimask at each point of the grid.

Default: false

Purchase credit to run this model

Each run costs approximately \$0.067. Alternatively, try out these models for free.

Manage billing

Output

Preview

JSON

output

output

Generated in

16.2 seconds

Tweak it

Iterate in playground

[Share](#)

[Download](#)

[Report](#)

[View full prediction](#)

[Show logs](#)

[Examples](#)

[View more examples](#)

[Run time and cost](#)

This model costs approximately \$0.067 to run on Replicate, or 14 runs per \$1, but this varies depending on your inputs. It is also open source and you can run it on your own computer with Docker.

To see how much you've spent, go to your dashboard.

This model runs on Nvidia L40S GPU hardware. Predictions typically complete within 69 seconds. The predict time for this model varies significantly based on the inputs.

[Readme](#)

Note: Currently, at this time this model only support image inputs (not video yet) and runs the large variant of the model.

[SAM 2: Segment Anything in Images and Videos](#)

[About](#)

Implementation of SAM 2, a model for segmenting objects in images and videos using various prompts.

[Limitations](#)

Performance may vary depending on image/video quality and complexity.

Very fast or complex motions in videos might be challenging.

Higher resolutions provide more detail but require more processing time.

SAM 2 is a model developed by Meta AI Research. It excels at segmenting objects in both images and videos with various types of prompts.

[Core Model](#)

[model architecture](#)

An overview of the SAM 2 framework.

SAM 2 uses a transformer architecture with streaming memory for real-time video processing. It builds on the original SAM model, extending its capabilities to video.

For more technical details, check out the Research paper.

Safety

Users should be aware of potential ethical implications: - Ensure you have the right to use input images and videos, especially those featuring identifiable individuals. - Be responsible about generated content to avoid potential misuse. - Be cautious about using copyrighted material as inputs without permission.

Support

All credit goes to the Meta AI Research team

Citation

```
@article{ravi2024sam2,  
  title={SAM 2: Segment Anything in Images and Videos},  
  author={Ravi, Nikhila and Gabeur, Valentin and Hu, Yuan-Ting and Hu, Rong-  
hang and Ryali, Chaitanya and Ma, Tengyu and Khedr, Haitham and R{a}dle,  
Roman and Rolland, Chloe and Gustafson, Laura and Mintun, Eric and Pan,  
Junting and Alwala, Kalyan Vasudev and Carion, Nicolas and Wu, Chao-Yuan  
and Girshick, Ross and Doll{a}r, Piotr and Feichtenhofer, Christoph},  
  journal={arXiv preprint},  
  year={2024}  
}
```

Copy

Model created over 1 year ago

Prompt 19

Use one of our client libraries to get started quickly.

Node.js

Python

HTTP

Set the REPLICATE_API_TOKEN environment variable

```
export REPLICATE_API_TOKEN=r8_G5H*****
```

Visibility

Copy

Learn more about authentication

Install Replicate's Node.js client library

```
npm install replicate
```

Copy

Learn more about setup

Run `lucataco/segment-anything-2` using Replicate's API. Check out the model's schema for an overview of inputs and outputs.

```
import { writeFile } from "fs/promises";
import Replicate from "replicate";
const replicate = new Replicate();
const input = {
  image: "https://replicate.delivery/pbxt/LMUHQoCnNzN15MpwWLPJUVF96g3zxer6p3dNTHtVrNxqMMe0/so
  mask_limit: 2
};
const output = await replicate.run("lucataco/segment-anything-2:be7cbde9fdf0eecdc8b20ffec9dd0d1cfeace0832
{ input });
// To access the file URLs:
console.log(output[0].url());
//=> "https://replicate.delivery/.../output_0.png"
// To write the files to disk:
for (const [index, item] of Object.entries(output)) {
  await writeFile(output_`${index}.png`, item);
}
//=> output_0.png, output_1.png written to disk
```

Prompt 20

```
aadarshbattula@Aadarshs-MacBook-Air server % npm run server
```

```
  aivideo-assessment@1.0.0 server
```

```
  node server/index.js
```

```
[dotenv@17.2.3] injecting env (1) from .env – tip:  enable debug logging with {
debug: true }
```

```
Server running on http://localhost:3001
```

Replicate token: configured

aadarshbattula@Aadarshs-MacBook-Air server %

Prompt 21

aadarshbattula@Aadarshs-MacBook-Air aivideo-assessment % lsof -i :3001

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

Google 681 aadarshbattula 40u IPv6 0x22b21e8108abce7d 0t0 TCP
localhost:55127->localhost:redwood-broker (ESTABLISHED)

node 62057 aadarshbattula 12u IPv6 0x22b21e8108aa867d 0t0 TCP *:redwood-
broker (LISTEN)

node 62057 aadarshbattula 17u IPv6 0x22b21e8108abde7d 0t0 TCP
localhost:redwood-broker->localhost:55127 (ESTABLISHED)

aadarshbattula@Aadarshs-MacBook-Air aivideo-assessment % npm run server

aivideo-assessment@1.0.0 server

node server/index.js

[dotenv@17.2.3] injecting env (1) from .env – tip: sync secrets across teammates
& machines: <https://dotenvx.com/ops>

Server running on http://localhost:3001

Replicate token: configured

aadarshbattula@Aadarshs-MacBook-Air aivideo-assessment %

Prompt 22

This error is still hitting: Error: Request to <https://api.replicate.com/v1/models/meta/sam-2-base/predictions> failed with status 404 Not Found: {"detail": "The requested resource could not be f

Prompt 23

[Request interrupted by user]

Prompt 24

Can you check if there is anyway to add debugging during the loading process to see what the model is doing when we send a request?

Prompt 25

The model was called and found 57 objects, but it is just highlighting the entire image pink. How would we implement hovering over each object would give a different color?

Prompt 26

Nothing is highlighting now. aadarshbattula@Aadarshs-MacBook-Air aivideo-assessment % npm run server

aivideo-assessment@1.0.0 server

node server/index.js

[dotenv@17.2.3] injecting env (1) from .env – tip: load multiple .env files with { path: ['.env.local', '.env'] }

Server running on http://localhost:3001

Replicate token: configured

=== Starting segmentation ===

Model: lucataco/segment-anything-2:be7cbde9fdf0eecdc8b20ffec9dd0d1cfeace0832d4d0b58a071d993182e1be0

Image length: 122087 chars

Image starts with: 

=== Segmentation complete ===

Output type: object

Output is array: true

Output length: 24

First item: ReadableStream { locked: false, state: 'readable', supportsBYOB: false }

Mask URL: URL {

href: 'https://replicate.delivery/xezq/K5WCyegUjt3MB6vVxqe0dw0kn2R7BLBlHXY09rCuZBQsIb9VA/mask'

origin: 'https://replicate.delivery',

```

protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/K5WCyegUjt3MB6vVxqe0dw0kn2R7BLBlHXY09rCuZBQsIb9VA/mask_0.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/uUeCumlW1M18Mya7uZtbXocdwzDBBvOI5ypPcl6t3UDWkteVA/mask_0.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/uUeCumlW1M18Mya7uZtbXocdwzDBBvOI5ypPcl6t3UDWkteVA/mask_1.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/wbAezDl3djQOQCWzAkl6fj8HGZn3hju5WXqedxkKbSYR26rA/mask_1.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',

```

```

host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/wbAezDl3djQOQCWzAkl6fj8HGZn3hJw5WXqxedxkKbSYR26rA/mask_2.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/ffkmB3Zps3uzGkeYbIz11fP66Ag4LiDEm4awQbUuqpPwis1XB/mask_3',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/ffkmB3Zps3uzGkeYbIz11fP66Ag4LiDEm4awQbUuqpPwis1XB/mask_3.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/cnQZGNqXxib0MZGp4OPnoVYygff6AlWAZ9TVtrRMBYBsIb9VA/ma',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',

```

```

pathname: '/xezq/cnQZGNqXxib0MZGp4OPnoVYygff6AlWAZ9TVtrRMBYBsIb9VA/mask_4.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/vL2OEMW2JhI8IN4XQxwyqlOohcAoJC5gwayADKKz18GLyWfKA/mask_4.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/vL2OEMW2JhI8IN4XQxwyqlOohcAoJC5gwayADKKz18GLyWfKA/mask_5.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/wn3moFXH5H6tGpsP93LwgF1XReC8UZu9DRuEHaWxDKYWkteVA/mask_5.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/wn3moFXH5H6tGpsP93LwgF1XReC8UZu9DRuEHaWxDKYWkteVA/mask_6.png',
search: '',
searchParams: URLSearchParams {},

```



```

hash: ""
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/wQ7CUijfrxSSTqRDxe7NoN0yYDa84fXjPGB2bvqGOKLdR26rA/mask_7.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: "",
password: "",
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: "",
pathname: '/xezq/wQ7CUijfrxSSTqRDxe7NoN0yYDa84fXjPGB2bvqGOKLdR26rA/mask_7.png',
search: "",
searchParams: URLSearchParams {},
hash: ""
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/0dvHISk9p7bFA5xeJeYqBr9Bz4SXTpw2ns5ZTHKcVHmuIb9VA/mask_8.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: "",
password: "",
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: "",
pathname: '/xezq/0dvHISk9p7bFA5xeJeYqBr9Bz4SXTpw2ns5ZTHKcVHmuIb9VA/mask_8.png',
search: "",
searchParams: URLSearchParams {},
hash: ""
}
Mask URL: URL {

```

```

href: 'https://replicate.delivery/xezq/yQhRFNjvAWI3PN2g921Z3lEzzkuMLBmUlJRTstNe9GEXkteVA/mask_
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/yQhRFNjvAWI3PN2g921Z3lEzzkuMLBmUlJRTstNe9GEXkteVA/mask_9.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/xIWroeeipupsgEqj5H403REBnnmPI3yIaX6y5Z8Cr4wuIb9VA/mask_10.
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/xIWroeeipupsgEqj5H403REBnnmPI3yIaX6y5Z8Cr4wuIb9VA/mask_10.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/eikuXcktNdWIbC3fDFw9tzRNLEl51RdspG02VkGtgoLuIb9VA/mask_1
origin: 'https://replicate.delivery',
protocol: 'https:',

```

```

username: "",
password: "",
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: "",
pathname: '/xezq/eikuXcktNdWIbC3fDFw9tzRNLEl51RdspG02VkGtgoLuIb9VA/mask__11.png',
search: "",
searchParams: URLSearchParams {},
hash: ""
}
Mask URL: URL {
  href: 'https://replicate.delivery/xezq/JvNaUWbRjMYDJ5VfQeQft8LEoADFFM1mwrElgFonXQQdR26rA/m',
  origin: 'https://replicate.delivery',
  protocol: 'https:',
  username: "",
  password: "",
  host: 'replicate.delivery',
  hostname: 'replicate.delivery',
  port: "",
  pathname: '/xezq/JvNaUWbRjMYDJ5VfQeQft8LEoADFFM1mwrElgFonXQQdR26rA/mask__12.png',
  search: "",
  searchParams: URLSearchParams {},
  hash: ""
}
Mask URL: URL {
  href: 'https://replicate.delivery/xezq/IW99NaPkWho2NxgzPPNfr78nFp3T8B1AZE4u2eORsXKvIb9VA/mask',
  origin: 'https://replicate.delivery',
  protocol: 'https:',
  username: "",
  password: "",
  host: 'replicate.delivery',

```

```

hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/IW99NaPkWho2NxgzPPNfr78nFp3T8B1AZE4u2eORsXKvIb9VA/mask_13.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/1rPEeMmi7bzIBytPtAvOBnt8VTyye4MsiPEvUAXFWneeis1XB/mask_13.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/1rPEeMmi7bzIBytPtAvOBnt8VTyye4MsiPEvUAXFWneeis1XB/mask_14.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/ptwSsfKwHRW8eUZO9pqav34hqLoo7rZxgUFY6ifyBsxeis1XB/mask_15.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/ptwSsfKwHRW8eUZO9pqav34hqLoo7rZxgUFY6ifyBsxeis1XB/mask_15.png',

```

```

search: "",
searchParams: URLSearchParams {},
hash: ""
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/ilh3e5an7iyelUZfrHxuKwkAJZHz7puwnfFesOyZz7F6FZrvC/mask_16.p',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: "",
password: "",
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: "",
pathname: '/xezq/ilh3e5an7iyelUZfrHxuKwkAJZHz7puwnfFesOyZz7F6FZrvC/mask_16.png',
search: "",
searchParams: URLSearchParams {},
hash: ""
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/tMY9m02FJj4GE1qANLFOKBgr3PrEvkilLRBGtpkXqK7LyWfKA/ma',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: "",
password: "",
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: "",
pathname: '/xezq/tMY9m02FJj4GE1qANLFOKBgr3PrEvkilLRBGtpkXqK7LyWfKA/mask_17.png',
search: "",
searchParams: URLSearchParams {},
hash: ""

```

```

}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/PvhESjlC9hb6GxgAivKVtEs6M4gxUZsOaCNeQr8zFefeis1XB/mask__18',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/PvhESjlC9hb6GxgAivKVtEs6M4gxUZsOaCNeQr8zFefeis1XB/mask__18.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/nIj6ahiWsnZbK9iBwSiA6TaglY6FhoIjEX5yLQCxvZ9LyWfKA/mask__19',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/nIj6ahiWsnZbK9iBwSiA6TaglY6FhoIjEX5yLQCxvZ9LyWfKA/mask__19.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/rY2yhA2qpSpYBJ5x029XSO1WGaM6zmWJcKlRIiCnlQeXkteVA/mask__20',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/rY2yhA2qpSpYBJ5x029XSO1WGaM6zmWJcKlRIiCnlQeXkteVA/mask__20.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}

```

```

origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/rY2yhA2qpSpYBJ5x029XSO1WGaM6zmWJcKIRIiCnlQeXkteVA/mask__20.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/eP9Pbhqy2vXmKSHg6XSlkODpoTwQxDlUrpvOGt49BAevIb9VA/mask__20.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/eP9Pbhqy2vXmKSHg6XSlkODpoTwQxDlUrpvOGt49BAevIb9VA/mask__21.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/CtfybauJ8eqjUEE2q5GBxhCZchsNc42w5XrlNDeLzK6fis1XB/mask__22.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',

```

```

password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/CtfybauJ8eqjUEE2q5GBxhCZchsNc42w5XrlNDeLzK6fis1XB/mask__22.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}
Mask URL: URL {
href: 'https://replicate.delivery/xezq/YLiBY7p04xKLEp4lReVXYu3Fd1RLOQXO8rtZ1RbRudtXkteVA/mask__22.png',
origin: 'https://replicate.delivery',
protocol: 'https:',
username: '',
password: '',
host: 'replicate.delivery',
hostname: 'replicate.delivery',
port: '',
pathname: '/xezq/YLiBY7p04xKLEp4lReVXYu3Fd1RLOQXO8rtZ1RbRudtXkteVA/mask__23.png',
search: '',
searchParams: URLSearchParams {},
hash: ''
}

```

Total masks: 24 What is the best way to go about debugging this?

Prompt 27

Received masks: 24

ImageEditor.jsx:40 === Loading masks ===

ImageEditor.jsx:41 Mask URLs received: (24) ['https://replicate.delivery/xezq/NXSh8nZsvNYhOdJ92Nik7yfb',
 'https://replicate.delivery/xezq/fdxsOHSibIz2cymcPfjO3CZcJ1UkqJ1J4vsoJ3xVA8dYLb9VA/mask__1.png',
 'https://replicate.delivery/xezq/8zdwOfmU9j2vRSOhMTFksofb1xwKxsOLsnfyWg1MaEVxW26rA/mask__2.p

'https://replicate.delivery/xezq/2YmILUFBLpqSO54pgLMO0SrSYfyP7UeyrjuIu8zVA17YLB9VA/mask_3.png',
'https://replicate.delivery/xezq/K1Pfd8K6nKwkF6n6JRwWddIqOLDSayNCJ3ZJA1Q64IQslteVA/mask_4.png',
'https://replicate.delivery/xezq/U3dBeIVMTvWPI6T25SPkhKhOj0ZPKCbByp6AnQWgteWYLB9VA/mask_5.png',
'https://replicate.delivery/xezq/KLeJHCg6bsR9PKS9QF3u6B0henekcLn8hXqurlmjVybwW26rA/mask_6.png',
'https://replicate.delivery/xezq/3tMZvem7uIWiaifp98CxRm5OAJebxdKm800dngTzCefHbZrvC/mask_7.png',
'https://replicate.delivery/xezq/eZ6lpldTE3xiSidmTYNAfOx9VrDjLBqpznzGLP1t3z9YLB9VA/mask_8.png',
'https://replicate.delivery/xezq/lGZOdj7LszoZLVbGWLlObLPsliDvZJs9ZLEjxQqseiBslteVA/mask_9.png',
'https://replicate.delivery/xezq/hOrNtsZaxz4yJReRfYQjIfUWGD9546Bb3yWPpCJeQNTnts1XB/mask_10.png',
'https://replicate.delivery/xezq/vYEIIsn9DYaYBR2Wec9kI8c9arTTl9eiPppCffbM1Eukts1XB/mask_11.png',
'https://replicate.delivery/xezq/C9jcth55HI4TJBrIZra8reBagU71ntWwxy2kh0kvRBsslteVA/mask_12.png',
'https://replicate.delivery/xezq/dkF6qlZNLso7EdFR685OU8KCe08aHshQcST8z6GWdqnslteVA/mask_13.png',
'https://replicate.delivery/xezq/eJMxJTbBeqhLfJKUYOBfnGjnCnZrPFwClp0VBBiSeDUObZrvC/mask_14.png',
'https://replicate.delivery/xezq/bVnQB2eX9FyjB664l7xl4OVvOO3fXmzEZ3ZuJnAsI8EZLb9VA/mask_15.png',
'https://replicate.delivery/xezq/B7d0XLBCcCbplxfqumbmteUt8ooBZmgvfIk9L5b8fh8lts1XB/mask_16.png',
'https://replicate.delivery/xezq/VfN7j5Wzkz3AFqJre1XOWp5qe7Comzbj1VX6UJCiBzNyW26rA/mask_17.png',
'https://replicate.delivery/xezq/GCHLTwVuiA6THJSSYiNZlhFcjUm8wORj69Wu9XHMhdb2yWfKA/mask_18.png',
'https://replicate.delivery/xezq/ASc8OLOMCi7RC1bf37MS0uQTQMihj9dXOKPPolsYNfQZLb9VA/mask_19.png',
'https://replicate.delivery/xezq/QNqJHIjmYlbfItrsfvBHymS08SDm8AKJayTtrcK7sjZLb9VA/mask_20.png',
'https://replicate.delivery/xezq/zpamIFrBR5anKNd3br0SzmNAbZbdHTFHivb8FsKfjyslteVA/mask_21.png',
'https://replicate.delivery/xezq/hDmbwMnUjwqZI1jz2LFZLSjaRehRFmWaB4j9uiloTTtslteVA/mask_22.png',
'https://replicate.delivery/xezq/fc4j0Xes5svNFkx9s69Rq4FCDFZMWxHvij13w5MXAaszW26rA/mask_23.png'

ImageEditor.jsx:46 Loading mask 0: https://replicate.delivery/xezq/NXSh8nZsvNYhOdJ92Nlk7yfbenbBM7nx

ImageEditor.jsx:46 Loading mask 1: https://replicate.delivery/xezq/fdxsOHSibIz2cymcPfjO3CZcJ1UkqJ1J4vs

ImageEditor.jsx:46 Loading mask 2: https://replicate.delivery/xezq/8zdwOfmU9j2vRSOhMTFksofb1xwKxsO

ImageEditor.jsx:46 Loading mask 3: https://replicate.delivery/xezq/2YmILUFBLpqSO54pgLMO0SrSYfyP7U

ImageEditor.jsx:46 Loading mask 4: https://replicate.delivery/xezq/K1Pfd8K6nKwkF6n6JRwWddIqOLDSayN

ImageEditor.jsx:46 Loading mask 5: https://replicate.delivery/xezq/U3dBeIVMTvWPI6T25SPkhKhOj0ZPKC

ImageEditor.jsx:46 Loading mask 6: https://replicate.delivery/xezq/KLeJHCg6bsR9PKS9QF3u6B0henekcLn8

ImageEditor.jsx:46 Loading mask 7: https://replicate.delivery/xezq/3tMZvem7uIWiaifp98CxRm5OAJebxdK

ImageEditor.jsx:46 Loading mask 8: https://replicate.delivery/xezq/eZ6lpldTE3xiSidmTYNAfOx9VrDjLBqpz

ImageEditor.jsx:46 Loading mask 9: https://replicate.delivery/xezq/lGZOdj7LszoZLVbGWLlObLPsliDvZJs9Z

ImageEditor.jsx:46 Loading mask 10: https://replicate.delivery/xezq/hOrNtsZaxz4yJReRfYQjIfUWGD9546B

ImageEditor.jsx:46 Loading mask 11: https://replicate.delivery/xezq/vYEIIsn9DYaYBR2Wec9kI8c9arTTl9eiP

ImageEditor.jsx:46 Loading mask 12: https://replicate.delivery/xezq/C9jcth55HI4TJBrIZra8reBagU71ntWwxy

ImageEditor.jsx:46 Loading mask 13: https://replicate.delivery/xezq/dkF6qlZNLso7EdFR685OU8KCe08aHsh

ImageEditor.jsx:46 Loading mask 14: https://replicate.delivery/xezq/eJMxJTbBeqhLfJKUYOBfnGjnCnZrPF

ImageEditor.jsx:46 Loading mask 15: https://replicate.delivery/xezq/bVnQB2eX9FyjB664l7xl4OVvOO3fXmzL

ImageEditor.jsx:46 Loading mask 16: <https://replicate.delivery/xezq/B7d0XLBCcCbPLxfqumbmteUt8ooBZm>

ImageEditor.jsx:46 Loading mask 17: <https://replicate.delivery/xezq/VfN7j5Wzkz3AFqJre1XOWp5qe7Comzb>

ImageEditor.jsx:46 Loading mask 18: <https://replicate.delivery/xezq/GCHLTwVuiA6THJSSYiNZlhFcjUm8wC>

ImageEditor.jsx:46 Loading mask 19: <https://replicate.delivery/xezq/ASc8OLOMCI7RC1bf37MS0uQTQMIHj>

ImageEditor.jsx:46 Loading mask 20: <https://replicate.delivery/xezq/QNqJHIjmYlbfItrsfvBHymS08SDm8AK>

ImageEditor.jsx:46 Loading mask 21: <https://replicate.delivery/xezq/zpamIFrBR5anKNd3br0SzmNAbZbdHT>

ImageEditor.jsx:46 Loading mask 22: <https://replicate.delivery/xezq/hDmbwMnUjwqZI1jz2LFZLSjaRehRFm>

ImageEditor.jsx:46 Loading mask 23: <https://replicate.delivery/xezq/fc4j0Xes5svNFkx9s69Rq4FCDFZMWxHv>

ImageEditor.jsx:50 Mask 0 loaded: 640x284

ImageEditor.jsx:60 Mask 0 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 2 loaded: 640x284

ImageEditor.jsx:60 Mask 2 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 3 loaded: 640x284

ImageEditor.jsx:60 Mask 3 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 8 loaded: 640x284

ImageEditor.jsx:60 Mask 8 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 5 loaded: 640x284

ImageEditor.jsx:60 Mask 5 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 1 loaded: 640x284

ImageEditor.jsx:60 Mask 1 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 4 loaded: 640x284

ImageEditor.jsx:60 Mask 4 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 7 loaded: 640x284

ImageEditor.jsx:60 Mask 7 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 14 loaded: 640x284

ImageEditor.jsx:60 Mask 14 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 9 loaded: 640x284

ImageEditor.jsx:60 Mask 9 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 10 loaded: 640x284

ImageEditor.jsx:60 Mask 10 area: 181760 pixels (100.0%)

ImageEditor.jsx:50 Mask 6 loaded: 640x284

ImageEditor.jsx:60 Mask 6 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 18 loaded: 640x284
ImageEditor.jsx:60 Mask 18 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 22 loaded: 640x284
ImageEditor.jsx:60 Mask 22 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 12 loaded: 640x284
ImageEditor.jsx:60 Mask 12 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 19 loaded: 640x284
ImageEditor.jsx:60 Mask 19 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 21 loaded: 640x284
ImageEditor.jsx:60 Mask 21 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 17 loaded: 640x284
ImageEditor.jsx:60 Mask 17 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 13 loaded: 640x284
ImageEditor.jsx:60 Mask 13 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 20 loaded: 640x284
ImageEditor.jsx:60 Mask 20 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 16 loaded: 640x284
ImageEditor.jsx:60 Mask 16 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 23 loaded: 640x284
ImageEditor.jsx:60 Mask 23 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 11 loaded: 640x284
ImageEditor.jsx:60 Mask 11 area: 181760 pixels (100.0%)
ImageEditor.jsx:50 Mask 15 loaded: 640x284
ImageEditor.jsx:60 Mask 15 area: 181760 pixels (100.0%)
ImageEditor.jsx:81 All masks loaded: 24
ImageEditor.jsx:89 After filtering (>80% removed): 0 masks
ImageEditor.jsx:97 Final masks with colors: []Final masks with colors: []length:
0[[Prototype]]: Array(0)

Prompt 28

Can you explain in depth what the error is?

Prompt 29

So I think the different objects are being recognized in the image and the image is changing different colors when hovered over. However, the the objects are supposed to be highlighted not the whole image. Use the attached image as reference

Prompt 30

[Image: source: /Users/aadarshbattula/Downloads/image.png]

Prompt 31

It is not highlighting the entire object. What are the thresholds we are using to determine what to highlight?

Prompt 32

Is there a way we can make this threshold tunable in the image editor?

Prompt 33

Mask 30 area: 19076 pixels (0.2%)

ImageEditor.jsx:58 Mask 30 loaded: 3600x2591

ImageEditor.jsx:68 Mask 30 area: 19076 pixels (0.2%)

ImageEditor.jsx:58 Mask 30 loaded: 3600x2591

ImageEditor.jsx:68 Mask 30 area: 19076 pixels (0.2%)

ImageEditor.jsx:58 Mask 30 loaded: 3600x2591

ImageEditor.jsx:68 Mask 30 area: 19076 pixels (0.2%)

ImageEditor.jsx:58 Mask 30 loaded: 3600x2591

ImageEditor.jsx:68 Mask 30 area: 19076 pixels (0.2%)

ImageEditor.jsx:58 Mask 30 loaded: 3600x2591

ImageEditor.jsx:64 Uncaught RangeError: Failed to execute 'getImageData' on
'CanvasRenderingContext2D': Out of memory at ImageData creation

at img.onload (ImageEditor.jsx:64:37) Is there a faster to change the threshold and load the

Prompt 34

Can you explain the entire current process from image upload to providing the masks?

Prompt 35

I dont see a difference in the highlighted image for the stricter mask and more inclusive mask.

Prompt 36

Can you review this commit and double check for any code not being used? Also double check that the code is readable and separated properly.

Prompt 37

Do a quick check that this MVP meets all the requirements. [Pasted text]

Prompt 38

[Request interrupted by user]

Prompt 39

Yes, just do steps 1 and 2 for now.

Prompt 40

This session is being continued from a previous conversation that ran out of context. The summary below covers the earlier portion of the conversation.

Analysis:

Let me chronologically analyze the conversation:

1. **Initial Context:** User shared information about AIVideo.com, a YC S23 company, and their interview process with Orestis Lykouropoulos. They had a technical assessment scheduled.
2. **Project Spec:** Orestis sent the task - build a standalone one-page image editor where hovering over objects/people shows a pixel mask selecting just that object. Requirements:
 - 1-2 hours, screen recorded
 - Open book (AI, internet allowed)
 - Must accomplish task AND be easy to use (UI/UX focus)
 - Production-scalable, commercially usable frameworks
3. **Technology Decision:** User chose Option 2 (Replicate API) for fastest implementation. Stack: React + Vite + Tailwind + Replicate's hosted SAM model.
4. **Implementation Steps:**
 - Project setup with React + Vite
 - Tailwind v4 configuration issues (needed @tailwindcss/postcss)
 - Created UploadZone component for drag & drop
 - Created ImageEditor component for display
 - Set up Express backend to hide API key
 - Integrated Replicate SAM 2 API
5. **Key Errors Encountered:**
 - Tailwind v4 PostCSS plugin issue
 - Wrong Replicate model name (meta/sam-2-base vs lucatoco/segment-anything-2)
 - URL objects vs strings from Replicate output
 - SAM masks are white-on-black, not transparent (alpha check failed)
 - Masks showing 100% coverage because checking alpha instead of RGB

- Memory issues with large images (3600x2591)
 - Threshold slider not working (SAM masks are binary)
6. **Color Palette:** User requested AIVideo.com colors:
 - Deep Black: #0D0D0D
 - Dark Gray: #1A1A1A
 - Hot Pink: #FF2D7C
 - White: #FFFFFF
 - Light Gray: #A0A0A0
 7. **Final State:** Working image segmentation with hover detection, opacity slider, mask overlay rendering.
 8. **Code Review Request:** User asked for review of unused code and readability. I identified:
 - Unused: `checkHealth()`, `containerRef`, unused properties in `generateColor`
 - Console logs to remove
 - Variable shadowing issue

User then asked to do steps 1 and 2 (remove unused code, remove console.logs) but then interrupted to request this summary.

Summary:

1. Primary Request and Intent:

The user (Aadarsh) was completing a technical assessment for an internship at AIVideo.com. The task from Orestis was to build a **standalone one-page image editor** where users can **hover over different objects/people and show a pixel mask selecting just that object**. Requirements included:

- 1-2 hour time limit, screen recorded
 - Open book (AI, internet, open source allowed)
 - Must accomplish the task AND have good UI/UX
 - Production-scalable, commercially usable frameworks
 - No labeling needed, just selection highlighting
2. Key Technical Concepts:
 - React 19 + Vite 7 for frontend
 - Tailwind CSS v4 (with @tailwindcss/postcss plugin)

- Express.js backend to proxy API calls (hide API key)
- Replicate API for hosting SAM 2 model
- SAM 2 (Segment Anything Model 2) by Meta for automatic mask generation
- Canvas API for pixel-level mask manipulation
- ImageData for reading/writing pixel values
- HSL to RGB color conversion for unique mask colors
- White-on-black mask format (not transparent PNGs)
- Base64 data URLs for image transfer

3. Files and Code Sections:

- **src/App.jsx** - Main orchestration component
 - Manages state: image, masks, isLoading, error
 - Calls segmentImage API on upload

```
import { useState } from 'react'

import UploadZone from './components/UploadZone'

import ImageEditor from './components/ImageEditor'

import { segmentImage } from './services/api'

function App() {

  const [image, setImage] = useState(null)

  const [masks, setMasks] = useState([])

  const [isLoading, setIsLoading] = useState(false)

  const [error, setError] = useState(null)

  // ... handleImageUpload, handleReset

}
```

- **src/components/UploadZone.jsx** - Drag & drop upload component

- Handles file selection, drag/drop, FileReader conversion to base64
- Clean, self-contained component (~70 lines)
- **src/components/ImageEditor.jsx** - Core display and interaction (~360 lines)
 - Most complex component with mask loading, processing, hover detection, rendering
 - Key functions:
 - * `calculateMaskArea(imageData, threshold)` - counts white pixels
 - * `generateColor(index, total)` - HSL colors via golden angle
 - * `getMaskAtPoint(x, y)` - hover detection
 - * HSL to RGB conversion for overlay rendering
 - Two-phase mask loading: `rawMasks` (loaded once) → `maskData` (filtered/processed)
 - Scales large images to max 1024px to prevent memory issues
- **src/services/api.js** - API client

```
const API_BASE = 'http://localhost:3001'
```

```
export async function segmentImage(imageDataUrl) { ... }
```

```
export async function checkHealth() { ... } // UNUSED
```

- **server/index.js** - Express backend
 - Proxies requests to Replicate API
 - Model: `lucataco/segment-anything-2:be7cbde9fdf0eecdc8b20ffec9dd0d1cfeace0832d4d`
 - Extracts URL strings from Replicate's URL objects
- **src/index.css** - Tailwind + AIVideo color variables

```
@import "tailwindcss";
```

```
:root {
```

```
  --color-deep-black: #0D0D0D;
```

```
  --color-dark-gray: #1A1A1A;
```

```

--color-hot-pink: #FF2D7C;

/* ... */

}

```

4. Errors and Fixes:

- **Tailwind v4 PostCSS error:** “tailwindcss directly as PostCSS plugin”
 - Fix: Installed `@tailwindcss/postcss`, updated `postcss.config.js`, changed CSS to `@import "tailwindcss"`
- **Replicate 404 error:** Wrong model name `meta/sam-2-base`
 - Fix: Changed to `lucataco/segment-anything-2` with correct version hash
- **URL objects instead of strings:** `item.url()` returns URL object
 - Fix: Extract `.href` property: `urlObj.href || urlObj.toString()`
- **Masks showing 100% area:** Checking alpha channel but SAM masks are opaque
 - Fix: Check RGB values instead: `r > threshold && g > threshold && b > threshold`
- **Entire image highlighted pink:** Using `source-in` compositing on opaque masks
 - Fix: Manually iterate pixels, set white→colored with alpha, black→transparent
- **Memory issues with large images:** 3600x2591 masks caused OOM
 - Fix: Scale down to max 1024px when loading, separate loading from processing
- **Threshold slider not visible effect:** SAM masks are binary (0 or 255)
 - Fix: Changed slider to control opacity instead of threshold

5. Problem Solving:

- Debugged API flow with extensive console logging
- Identified SAM mask format (white-on-black, not transparent)
- Optimized mask loading (load once, process on threshold change)
- Added image downscaling for memory management

- Implemented golden angle color distribution for unique mask colors
6. All User Messages:
- “Browse startups...” (shared AIVideo.com context and email thread)
 - “understand this:” (shared job posting details)
 - “No, he will send me the project description in a couple minutes”
 - Shared Orestis’s project spec with task details
 - “Lets go ahead with option 2. Can you plan out the steps need to implement this?”
 - “Can you explain what replicate.com is?”
 - “I created an API key for replicate in the .env... Lets proceed with the basic react app setup”
 - “I think we need to install tailwind” (error message)
 - “Ok this looks fine for now. Is the react app set up finished?”
 - “Yes proceed with this step”
 - “Can you change it to follow this color palette?” (AIVideo colors)
 - “Can you center the upload different image button?”
 - “Also, can you break up App.jsx into multiple files”
 - “Yes, lets move to that setup” (Replicate API)
 - “Can you say the command to run the backend server”
 - “Error: Request to <https://api.replicate.com/v1/models/meta/sam-2-base/predictions> failed”
 - Shared Replicate documentation and correct model info
 - “Can you provide more options and outline the key benefits and issues?”
 - “This error is still hitting... Can you check if there is anyway to add debugging”
 - “The model was called and found 57 objects, but it is just highlighting the entire image pink”
 - “Nothing is highlighting now” (after filtering fix)
 - “Can you explain in depth what the error is?”
 - “So I think the different objects are being recognized... However, the objects are supposed to be highlighted not the whole image” (shared reference image)

- “It is not highlighting the entire object. What are the thresholds we are using?”
- “Is there a way we can make this threshold tunable in the image editor?”
- Shared memory error logs
- “Is there a faster way to change the threshold and load the new image”
- “Can you explain the entire current process from image upload to providing the masks?”
- “I dont see a difference in the highlighted image for the stricter mask and more inclusive mask”
- “Can you review this commit and double check for any code not being used?”
- “Yes, just do steps 1 and 2 for now” (then interrupted for summary)

7. Pending Tasks:

- Remove unused code (`checkHealth()`, `containerRef`, unused `generateColor` properties)
- Remove `console.log` statements from all files
- (Not yet requested) Extract HSL→RGB conversion to utility function
- (Not yet requested) Fix variable shadowing (`maskData` used twice)

8. Current Work:

User asked for a code review, I identified issues, and user said:

“Yes, just do steps 1 and 2 for now”

Steps 1 and 2 from my review were:

1. Remove unused code
2. Remove `console.log`s

User then interrupted to request this summary before I could implement those changes.

9. Optional Next Step:

Based on the user’s last explicit request: **“Yes, just do steps 1 and 2 for now”**

The next step is to:

1. Remove unused code:
 - Delete `checkHealth()` from `src/services/api.js`

- Remove `containerRef` from `src/components/ImageEditor.jsx`
 - Clean up unused properties in `generateColor()`
2. Remove `console.log` statements from:
- `server/index.js` (lines 34-37, 59-63, 76, 80)
 - `src/App.jsx` (line 21)
 - `src/components/ImageEditor.jsx` (lines 51, 93, 107, 128)

If you need specific details from before compaction (like exact code snippets, error messages, or content you generated), read the full transcript at: `/Users/aadarshbattula/.claude/projects/-Users-aadarshbattula-Projects-aivideo-assessment/63a7609b-1b78-41b7-8d98-2c10eb796483.jsonl`

Please continue the conversation from where we left it off without asking the user any further questions. Continue with the last task that you were asked to work on.

Prompt 41

Is any filtering being done in the code when the model sends back the image urls with areas?

Prompt 42

Can you spend some time writing some documentation in the readme. Be concise, dont use emojis, and simply explain what this does. Dont add licensing or go too indepth.

Prompt 43

can you provide me a pdf with all the prompts I submitted to you in this session? Do you have that information or do you lose it when you compact?