

DEEP LOOKING

A User Guide

Paul Fishwick and Claude Code

February 2026

Contents

1	Introduction	2
2	Getting Started	2
2.1	Opening the Application	2
2.2	The Gallery	2
3	The Viewer	3
3.1	Segment Tooltip	3
3.2	Clicking on Regions	3
3.3	Multiple Simultaneous Popups	3
3.4	Lens Tabs	3
3.5	Popup Interaction	4
3.6	Image Lightbox	5
3.7	Whole-Image Lenses	5
4	Content Structure	5
4.1	Directory Hierarchy	5
4.2	Gallery Configuration	6
4.3	Manifest File	6
4.4	Markdown Files	7
4.5	Segmentation Masks	8
5	Adding New Content	9
5.1	Adding a New Artwork	9
5.2	Adding a New Lens to an Existing Segment	9
5.3	Adding a Whole-Image Lens	9
6	Current Artworks	9
7	Technical Notes	9

1 Introduction

Deep Looking is an interactive web application for exploring visual art through discipline-specific *lenses*. Rather than viewing an artwork passively, visitors click on regions of a painting to discover interpretive content drawn from poetry, mathematics, computer science, geology, botany, material science, physics, language, and cinematography. Multiple popups can be open simultaneously, allowing side-by-side comparison of different regions and lenses.

The application runs entirely in the browser as a single `index.html` file with no server-side code, making it compatible with GitHub Pages. All content is loaded dynamically from JSON manifests and Markdown files organized in a structured directory hierarchy.

2 Getting Started

2.1 Opening the Application

Open `index.html` in a web browser, or serve it via a local HTTP server:

```
python3 -m http.server 8765
```

Then navigate to `http://localhost:8765/index.html`.

2.2 The Gallery

The application opens with a gallery of artwork thumbnails, as shown in Figure 1. Each card displays the painting, its title, the artist, and the hosting gallery or museum (linked to the institution's website). Click any card to open that artwork in the interactive viewer.

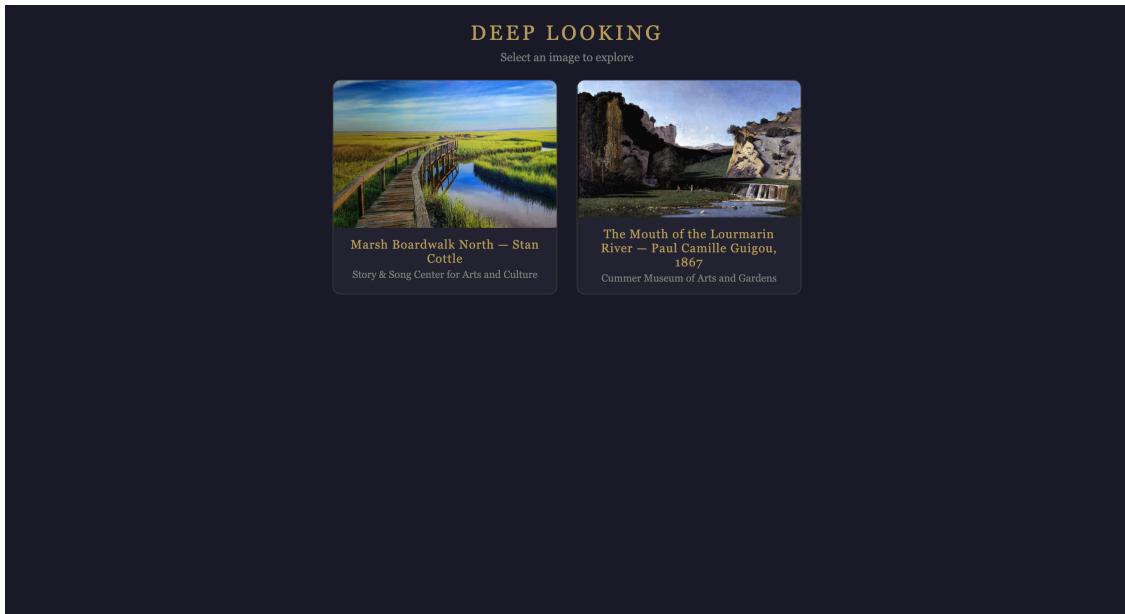


Figure 1: The gallery screen showing two artworks: *Marsh Boardwalk North* by Stan Cottle and *The Mouth of the Lourmarin River* by Paul Camille Guigou (1867).

3 The Viewer

Clicking an artwork opens the viewer, as shown in Figure 2. The viewer displays the full artwork image with a crosshair cursor, indicating that the image is interactive. A “Back to Gallery” button in the upper left returns to the gallery screen.

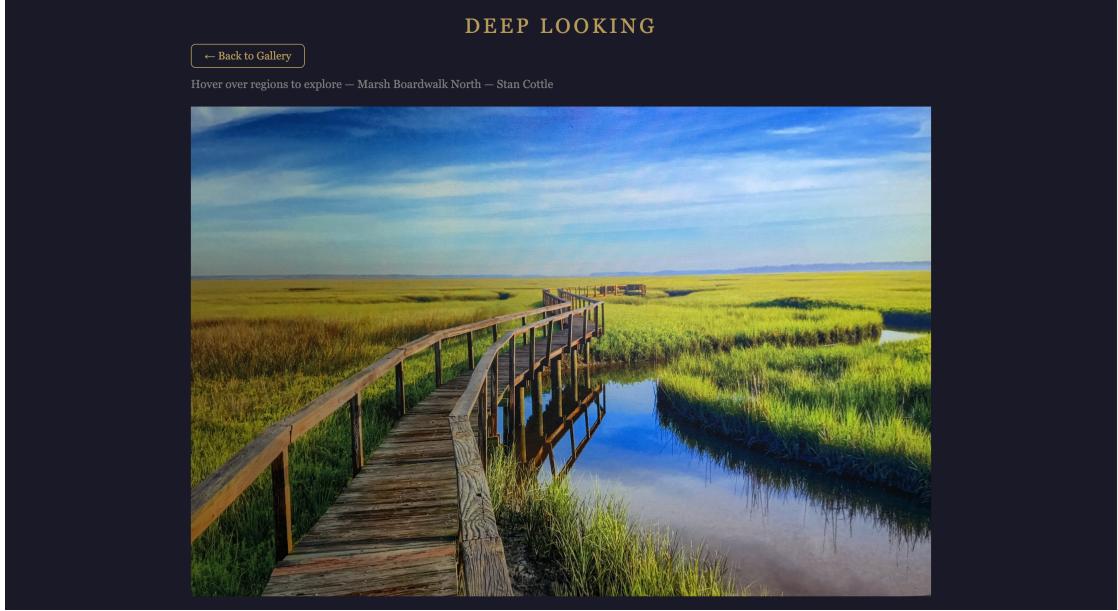


Figure 2: The viewer showing *Marsh Boardwalk North*. The crosshair cursor invites the visitor to explore regions of the painting.

3.1 Segment Tooltip

As the visitor moves the cursor over the artwork, a small tooltip appears showing the name of the region under the cursor (e.g., “Sky”, “Boardwalk”, “Tidal Creek Water”). This provides orientation before clicking to explore a region’s content.

3.2 Clicking on Regions

Clicking on a recognized region of the artwork opens a popup showing interpretive content through a lens, as shown in Figure 3. Each click creates a new popup positioned near the click location.

3.3 Multiple Simultaneous Popups

Clicking on different regions opens additional popups without closing existing ones. This allows visitors to compare content across regions side by side. Each popup operates independently — its tabs can be switched, it can be dragged and resized, and it is closed individually via its X button. Returning to the gallery clears all open popups.

3.4 Lens Tabs

When a region has multiple lenses, a tab bar appears at the top of the popup (see Figure 3). Clicking a tab switches the popup content to that lens. The available lens types are listed in Table 1.

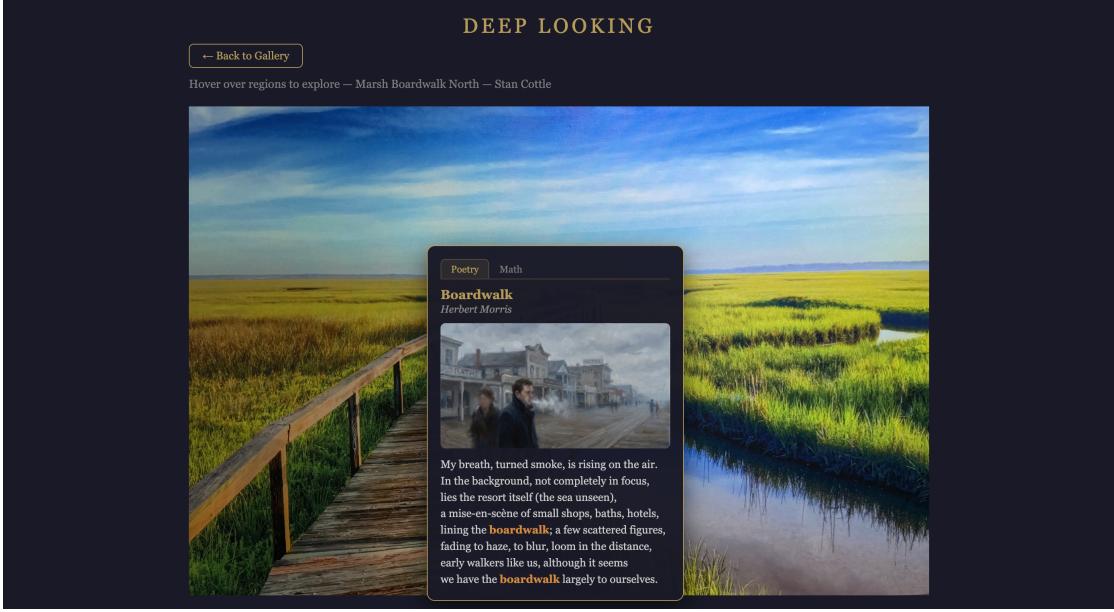


Figure 3: A popup opened by clicking on the boardwalk region in the Marsh painting. The popup shows the Poetry lens with a poem excerpt by Herbert Morris. Note the Poetry and Math tabs at the top, indicating two available lenses for this region.

When a region has only a single lens, the tab still appears as a label. The tab bar and the popup's border/padding areas serve as drag handles (see Section 3.5).

Lens	Tab Label	Content Type
Poetry	Poetry	Poem excerpt with highlighted keyword
Mathematics	Math	Geometric and mathematical analysis
Computer Science	Comp Sci	Knowledge graphs, semantic networks
Language	Language	Etymology and linguistic analysis
Physics	Physics	Physical phenomena (e.g., Rayleigh scattering)
Geology	Geology	Rock formations, geological processes
Botany	Botany	Flora identification, ecosystems
Material Science	Material Sci	Pigments, paint chemistry
Cinematography	Cinema	Video bringing the scene to life

Table 1: The nine lens types currently available in Deep Looking.

3.5 Popup Interaction

Every popup created by clicking on a region is immediately interactive, as shown in Figure 4. Each popup has the following properties:

- A close button (X) appears in the top-right corner
- The title is a hyperlink to a reference source (e.g., the full poem, a Wikipedia article)
- The popup can be **dragged** by clicking and dragging the tab bar or any border/padding area (indicated by a grab cursor)
- The popup can be **resized** from any of its four corners

Clicking the X button closes that individual popup. Multiple popups can remain open simultaneously.

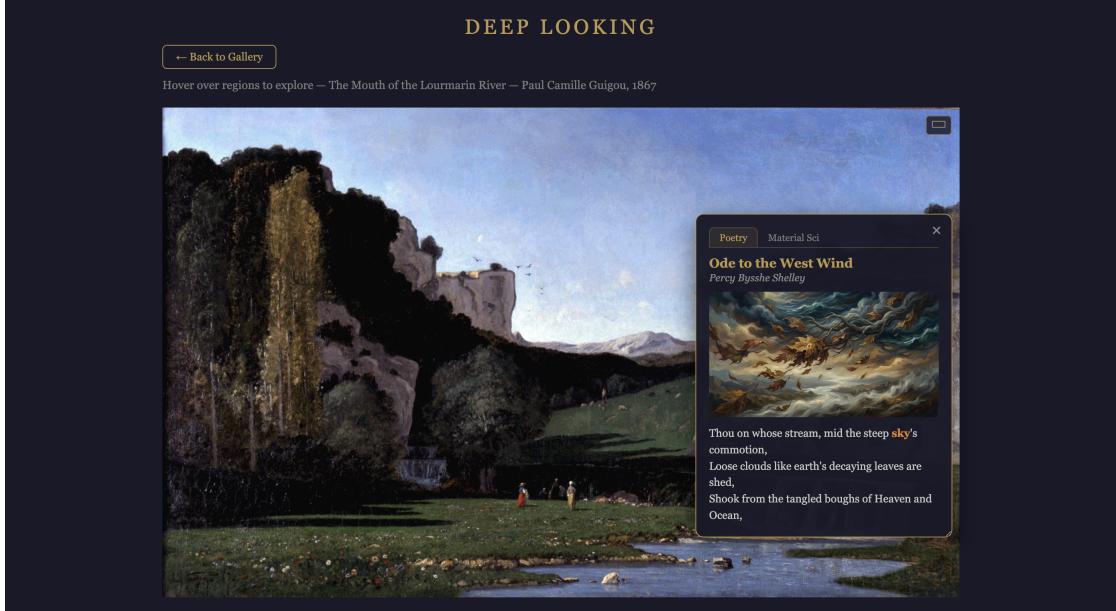


Figure 4: A pinned popup on the Lourmarin painting showing *Ode to the West Wind* by Percy Bysshe Shelley. The Poetry and Material Sci tabs are visible, the close button appears, and the title is now a clickable hyperlink. The keyword “sky” is highlighted in orange.

3.6 Image Lightbox

When a popup contains an image, clicking on that image opens a full-screen lightbox overlay (Figure 5). The lightbox displays the image at full size against a dark background. Clicking anywhere in the lightbox dismisses it.

3.7 Whole-Image Lenses

Some lenses apply to the entire artwork rather than a specific region. When whole-image lenses are available, a small rectangular icon button appears in the upper-right corner of the artwork (Figure 6). Clicking this button opens a pinned popup with the whole-image lens content.

Currently, cinematography is the only whole-image lens. It displays a video that imagines the static painting coming to life. Clicking on the video in the popup shows it in a full-screen lightbox, just like images (Figure 7).

4 Content Structure

4.1 Directory Hierarchy

Deep Looking uses a structured directory hierarchy that mirrors the conceptual model of gallery, image, segment, and lens:

```
deeplooking/  
    index.html
```

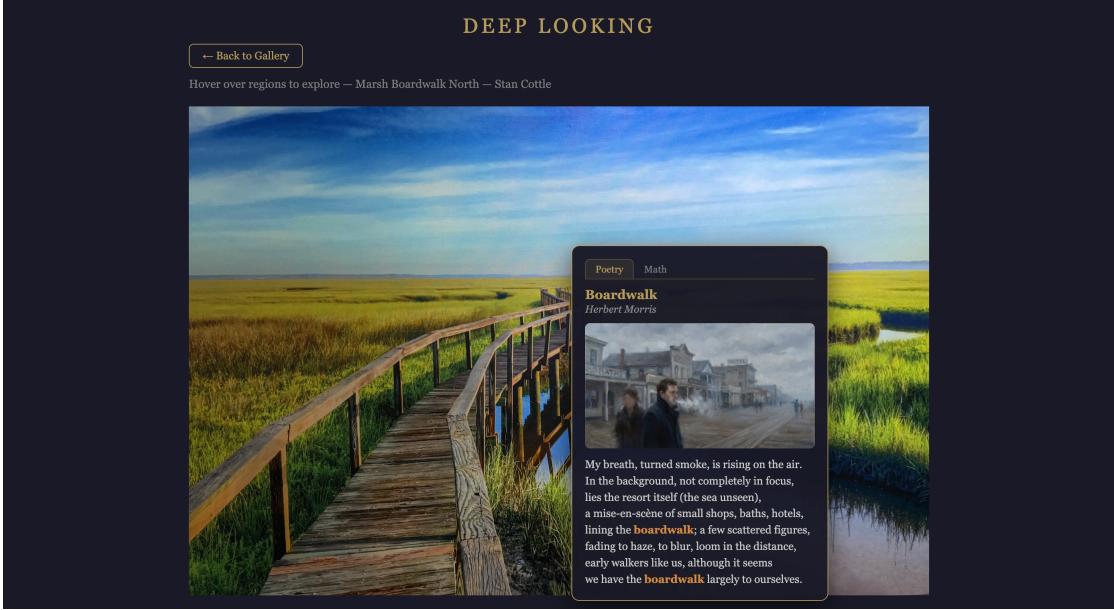


Figure 5: The image lightbox, activated by clicking on the illustration in a popup. The boardwalk illustration is shown at full size. Clicking anywhere dismisses the overlay.

```

images/
  gallery.json
  <image-id>/
    manifest.json
    <artwork-file>
      segmentation_mask.png
      <segment>/
        <lens>/
          <segment>.md
          <image-or-video-file>
    wholeimage/
      <lens>/
        wholeimage.md
        <video-file>

```

4.2 Gallery Configuration

The file `images/gallery.json` is a JSON array of image directory names:

```
[ "marsh", "lourmarin" ]
```

The application loads each image's manifest and builds the gallery grid dynamically.

4.3 Manifest File

Each image directory contains a `manifest.json` that defines the artwork metadata, segmentation mask, and the mapping of segments to lenses:

```
{
  "id": "marsh",
```

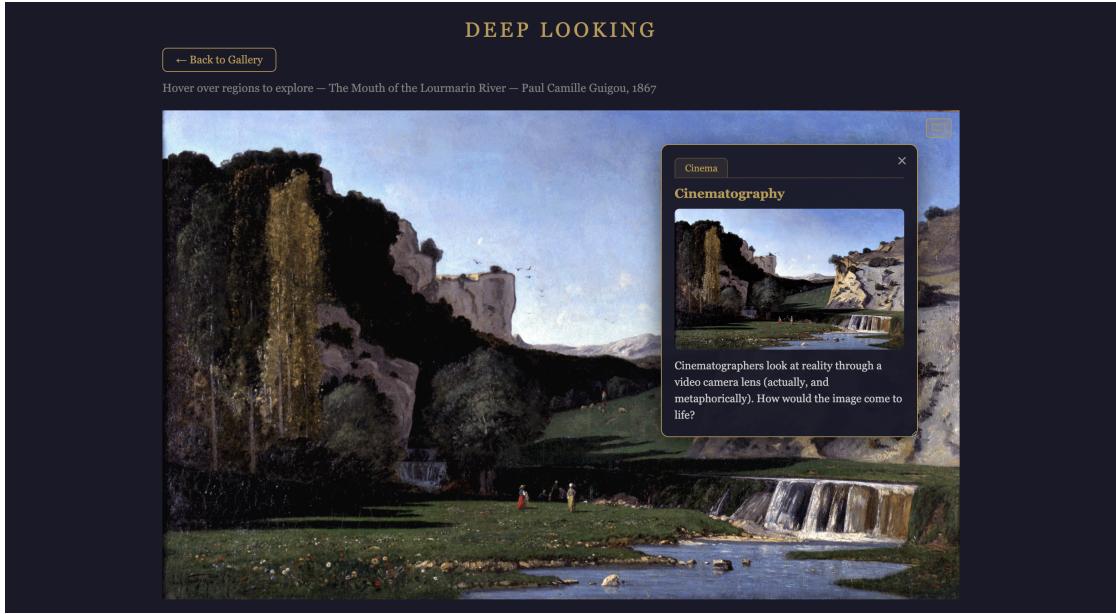


Figure 6: The whole-image cinematography lens on the Lourmarin painting. The rectangular icon button is visible in the upper-right corner. The popup shows a video with the Cinema tab and a description of the cinematographic interpretation.

```

"title": "Marsh Boardwalk North -- Stan Cottle",
"gallery": "Story & Song Center for Arts and Culture",
"galleryUrl": "https://storyandsongarts.org/",
"image": "marsh.jpeg",
"mask": "segmentation_mask.png",
"imageLenses": ["cinematography"],
"segments": [
  {
    "label": "Boardwalk",
    "dir": "boardwalk",
    "lenses": ["poetry", "math"],
    "colorRule": { "r_gt": 110, "g_lt": 110, "b_lt": 60 }
  }
]
}

```

The `colorRule` object defines RGB threshold conditions that map segmentation mask pixel colors to segment labels. The `imageLenses` field (optional) lists lenses that apply to the whole image, stored under the `wholeimage/` directory.

4.4 Markdown Files

Each lens provides its content via a Markdown file with a simple front-matter format:

```

title: Boardwalk
poet: Herbert Morris
url: https://newcriterion.com/article/boardwalk/
image: marsh_boardwalk.png
keyword: boardwalk

```



Figure 7: The video lightbox, activated by clicking on the video in the cinematography popup. The video plays at full size against a dark background.

My breath, turned smoke, is rising on the air.
In the background, not completely in focus,
lies the resort itself (the sea unseen),
a mise-en-scene of small shops, baths, hotels,
lining the boardwalk...

Fields:

- `title` — displayed as the popup heading (required)
- `poet` — author or attribution line (may be empty)
- `url` — hyperlink on the popup title (may be empty)
- `image` — filename of an associated image in the same directory (may be empty)
- `video` — filename of a video file, used instead of `image` (may be empty)
- `keyword` — word highlighted in orange within the body text (may be empty)

The body text (everything after the blank line) is displayed in the popup below the image or video.

4.5 Segmentation Masks

Each artwork has an associated `segmentation_mask.png` — a flat-color image where each region is painted a distinct solid color. These masks are generated using Google Gemini (Nano Banana Pro) or similar segmentation tools. The application reads mask pixels via a hidden HTML canvas element to determine which segment the cursor is over when clicked.

5 Adding New Content

5.1 Adding a New Artwork

1. Create a directory under `images/` with a unique ID (e.g., `images/mypainting/`)
2. Place the artwork image and segmentation mask in this directory
3. Create a `manifest.json` defining the title, image, mask, and segments with color rules
4. Add the directory name to `images/gallery.json`
5. Create subdirectories for each segment, then lens subdirectories within each
6. Add Markdown files following the front-matter format described above

5.2 Adding a New Lens to an Existing Segment

1. Create a subdirectory under the segment directory with the lens name
2. Add a Markdown file named `<segment>.md` with front-matter and body text
3. Optionally add an image or video file referenced in the Markdown
4. Add the lens name to the segment's `lenses` array in `manifest.json`

5.3 Adding a Whole-Image Lens

1. Create the directory `<image>/wholeimage/<lens-name>/`
2. Add a Markdown file named `wholeimage.md` with front-matter and body text
3. Add the lens name to the `imageLenses` array in `manifest.json`

6 Current Artworks

The application currently includes two artworks, summarized in Table 2.

Artwork	Segment	Lenses	Whole-Image
Marsh Boardwalk North (Stan Cottle)	Boardwalk Salt Marsh Vegetation Sky Tidal Creek Water	Poetry, Math Poetry, Comp Sci, Language Poetry, Physics Poetry	—
The Mouth of the Lourmarin River (P. C. Guigou, 1867)	Sky Cliffs Meadow River Figures Trees	Poetry, Material Sci Poetry, Geology Poetry, Botany Poetry Poetry Poetry	Cinema

Table 2: Current artworks with their segments and available lenses.

7 Technical Notes

- The application is a single HTML file with embedded CSS and JavaScript — no build tools or frameworks are required.
- Segmentation detection uses pixel-level color matching on a hidden canvas, avoiding the need for polygon data.

- Proportional coordinate mapping handles the mismatch between the displayed image size and the mask resolution.
- All content is loaded dynamically via `fetch()`, with cache-busting query strings to ensure fresh data during development.
- The application is fully compatible with static hosting (e.g., GitHub Pages).