

Captcha-er Chrome Extension - Documentation

Project Overview:

This Chrome extension automates the process of solving captcha images on websites. It uses image processing techniques to recognize and solve captchas efficiently. The extension operates by extracting the captcha image from the page, processing it, and then filling the captcha solution into the appropriate field.

Key Components:

1. **manifest.json**
The configuration file that defines the properties and permissions of the Chrome extension. It specifies the extension's name, description, version, permissions, and the content scripts to be injected into the web page.
 2. **content.js**
The core script responsible for processing and solving captchas. It uses the image data, processes it, and identifies the characters in the captcha image. It then auto-fills the captcha field with the recognized characters.
 3. **bitmaps.js**
A supporting script that contains predefined bitmaps (character patterns) and their corresponding weights and biases for the captcha solver's recognition mechanism. This script is used for comparing parts of the captcha image with known patterns.
 4. **background.js**
The background script that handles any background tasks, such as initialization or message passing, though it's not strictly necessary for this specific implementation.
 5. **popup.html**
A simple popup interface for the user, which is optional. If implemented, this file would be displayed when the user interacts with the extension icon in the browser.
-

Features:

- **Captcha Recognition:**
The extension uses image processing techniques to recognize captcha characters in an image. It scans the image, compares it with known character patterns, and outputs the recognized characters.
- **Auto-fill:**
The extension automatically fills in the captcha input field with the correct answer once the captcha is recognized.

- **Automatic Refresh:**

The extension listens for captcha refresh events, automatically solving and filling the captcha when the page or captcha refreshes.

How It Works:

1. **Injecting Scripts:**

When the user visits a page with a captcha image (e.g., <https://vtopcc.vit.ac.in/vtop/initialProcess>), the extension injects `bitmaps.js` and `content.js` into the page.

2. **Captcha Image Detection:**

The script detects captcha images on the page by looking for an `img` tag with a specific `alt` attribute (e.g., `vtopCaptcha`).

3. **Image Processing:**

Once the captcha image is identified, `content.js` uses `uri_to_img_data()` to convert the image into an array of pixel data. It processes this data using the `captcha_parse()` function to recognize the characters in the captcha image.

4. **Recognition Algorithm:**

The recognized characters are determined by comparing parts of the image with the pre-defined bitmaps (from `bitmaps.js`). This matching process involves checking each possible character against a list of predefined characters.

5. **Auto-fill Captcha Field:**

After recognizing the captcha, the script automatically fills the captcha field with the correct characters.

6. **Captcha Refresh Handling:**

The extension monitors for captcha refreshes and automatically solves and fills the new captcha when the page reloads or the captcha changes.

Installation:

1. **Clone the Repository:**

Download or clone the repository containing the extension files.

2. **Load the Extension in Chrome:**

- Open Chrome and navigate to `chrome://extensions/`.
- Enable **Developer mode** in the top-right corner.
- Click on **Load unpacked** and select the folder containing the extension files.

3. **Enable the Extension:** Once the extension is installed, it will be active in the background and will begin solving captchas on compatible pages.

Dependencies:

- **No external dependencies** are required for this extension. It uses native JavaScript APIs to process the captcha image and perform the necessary image recognition tasks.
-

Permissions:

- **activeTab**: This permission allows the extension to interact with the current active tab.
 - **storage**: This permission is used to store any necessary data, such as user preferences or settings, within the browser.
-

Troubleshooting:

- **Captcha not solved correctly:**
Ensure that the captcha image is properly detected on the page. You may need to adjust the detection logic if the image's structure or format is different from the one the extension expects.
 - **Extension not working on all websites:**
The extension is configured to work on all pages (http://**/* and https://**/*), but specific captchas might require adjustments to the recognition algorithm. If you encounter issues, try updating the `bitmaps.js` file with the new captcha patterns.
-

Future Improvements:

- **Enhancing Recognition Accuracy:**
The captcha recognition algorithm can be further optimized to handle more complex captchas with better accuracy.
- **Multi-language Support:**
Currently, the extension supports characters in a specific set. Adding support for different languages and character sets could make the extension more versatile.
- **Error Handling and Logging:**
Implementing more robust error handling and logging mechanisms will help improve the extension's reliability and user experience.