Automating Sticker Generation With Web Scraping and Image Processing in Python

Quickly generate stickers for messaging platforms

Danil Vityazev Follow
Oct 26 · 3 min read

Photo by Slidebean on Unsplash

In this article, I'm going to tell you about how I generated 42 Telegram stickers out of images of posters from an online shop. The shop sells various posters with funny puns, but there are no corresponding stickers. Let's create them!

The only problem is that to make a single sticker one needs to download a picture from the web page, separate the letters from the background in photoshop and save in the appropriate resolutions for telegram stickers. This would be extremely time-consuming for 42 images, so let's try to automate the process.

So here's the plan:

1. Web scrapping — parse all the pictures from the online shop.
2. Automatically separate letters from the background, delete shades from the background and make pictures look more like scans.

Downloading Pictures

First of all, let's create a list of all the links to the pages of individual posters. This way, we'll be able to download high-res pictures. The resolution of the pictures in the main gallery is too small.

In order to do that, let’s save all the links from the pictures in the gallery.

The gallery is divided into four pages, each containing 15 posters. To get to the next page it’s enough to add `?offset=` 15 or 30 or 45. This allows us to parse the remaining links.

The last part is to follow all the links and download high-res pictures from the product pages.

Pictures are named by the text on the posters, so we need to delete forbidden characters from the name like “/”, “?” and “*”.

Image Processing

Unfortunately, the online shop contains photos of the posters instead of scans. So we cannot use them as-is. Here’s an example of how a typical picture looks:

The poser on the picture is lit from the side. This makes the texture of the paper clearly visible. In some cases, it’s challenging to separate letters using a brightness threshold, because under certain circumstances letters can be brighter than some parts of the background. This approach is not universal in this case.

To implement an approach that would suit all the possible light conditions, let’s use the fact that all the posters contain big red letters on the white or gray background. Each pixel of a digital image is a vector of three numbers. A gray pixel, no matter if it is bright or dim, consists of three similar numbers, while a pixel of color will contain numbers that are different from each other.

So it’s way more effective to separate a background based on standard deviation threshold rather than brightness. After several tries, I figured out that a good estimate for such a threshold would be 30.

After detecting a pixel as being a part of the background, let’s set its value to (245, 245, 245). Pixels that are part of letters will be set closer to (200, 17, 11). The background will not be perfectly white, but this looks better as there is no perfect white paper, and the sticker set needs to imitate posters.

Here’s the result:

Now the last step is to resize the image so that the longest dimension is 512. To resize all the images and save them as png files, I'll use the PIL library.

Here's the result:

And the link to GitHub for those who are interested.

Thanks to Anupam Chugh.

Programming Python Image Processing Data Science Machine Learning