C2C pattern generation from image

Anja Brelih

Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana

Abstract. This paper describes a project that was made for second seminar in course Interaction and Information design. Selected project generates a pattern from uploaded image with Python Imaging Library by reducing image size and colors. Project was then integrated into a web application with Streamlit where the user can interactively change pattern parameters.

Keywords: Python · Streamlit · Pillow · Image processing.

1 Introduction

In this paper I present my methods in creating a web application for C2C pattern generation from image. C2C patterns are instructions used in corner-to-corner (C2C) crochet technique and can be quite expensive to buy.

Paper presents my inspiration for this project, tools I used for my solution, and describes the required steps for image processing to generate a desired pattern with example images.

2 Inspiration and related work

At the beginning of the of the semester I faced a problem for my personal crochet project. I wanted to make a crochet blanket with a C2C technique but I could not find the appropriate pattern. At the same time we started lectures on Interaction and Information Design course, where we saw different projects in New media art, Visualization and Interactivity which gave me the idea to create an application that can create a C2C pattern from image.

When I was searching for similar project I discovered a few web applications that offer the same functionality as I need, but none of those provided pattern for free. My idea is, that when my application becomes publicly available, anyone can create patterns for free.

3 Tools and libraries

Application is realized in programming language Python with library Pillow, where I did proof of concept for pattern generation. Later I integrated my code with Streamlit and it can be deployed as a web application.

3.1 Python Imaging Library

Python Imaging Library (Pillow) adds image processing capabilities to Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities [1].

Pillow is used for all image processing in this project. Used modules are *Image*, for resizing and converting, *ImageOps* for expanding image and *ImageDraw* for drawing lines and text.

3.2 Streamlit

Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app [3].

Although Streamlit is originally not meant for image processing applications, and it is a fairly new tool which is still developing, I chose it for my application. At the end I integrated all the functionalities to the web application, but I had to overcome a few of the missing capabilities of Streamlit.

4 Application

User uploads an image in one of the standard formats (*.jpg*, *.jpeg*, *.png*) and reads its size (see Fig. 1).

Fig. 1. Original image example.

4.1 Image resize

First step is to reduce the number of pixels in the image. This is done by resizing the image a new size which is specified by user in the application. This input value is then the width of the pattern or in other words, number of squares along X axis. After first resizing, we need to resize the image to a standard pattern size (width of A4 paper). Both resizing are done with filter *NEAREST* which picks the nearest pixel from the input image (see Fig. 2) [1].

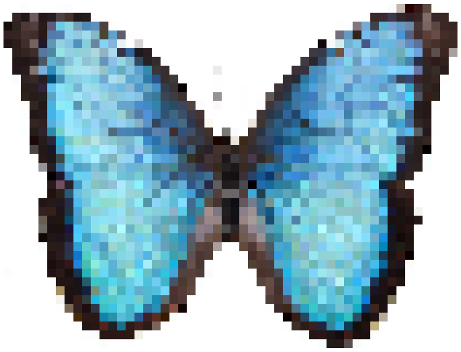


Fig. 2. Reduced number of pixels example.

4.2 Color reduction

In second step we need to reduce the number of colors on the image in order to create a pattern. Number of colors used is specified by user. For this step we use convert function which converts RGB image to P with adaptive palette, in which the colors are selected from the original image by picking the most frequent colors (see Fig. 3).

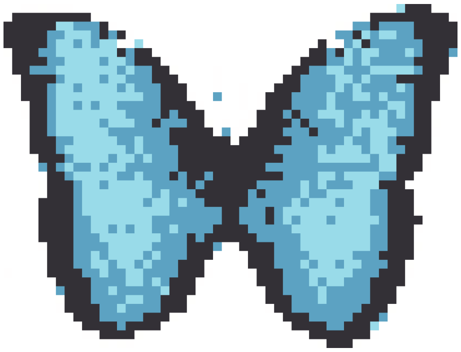


Fig. 3. Reduced number of colors example.

4.3 Draw grid and numbers

The last step is to draw a grid on the image to separate the squares and print numbers on all sides so it is easier to read the pattern. Both are done with *ImageDraw* module.

Grid is draw with *line* on *X* and *Y* axis with for loop, and current width and and height and desired width and scaled height. Same principle is used for drawing numbers, we just use different coordinates and use function *text* with iteration of the loop as string value (see Fig. 4).

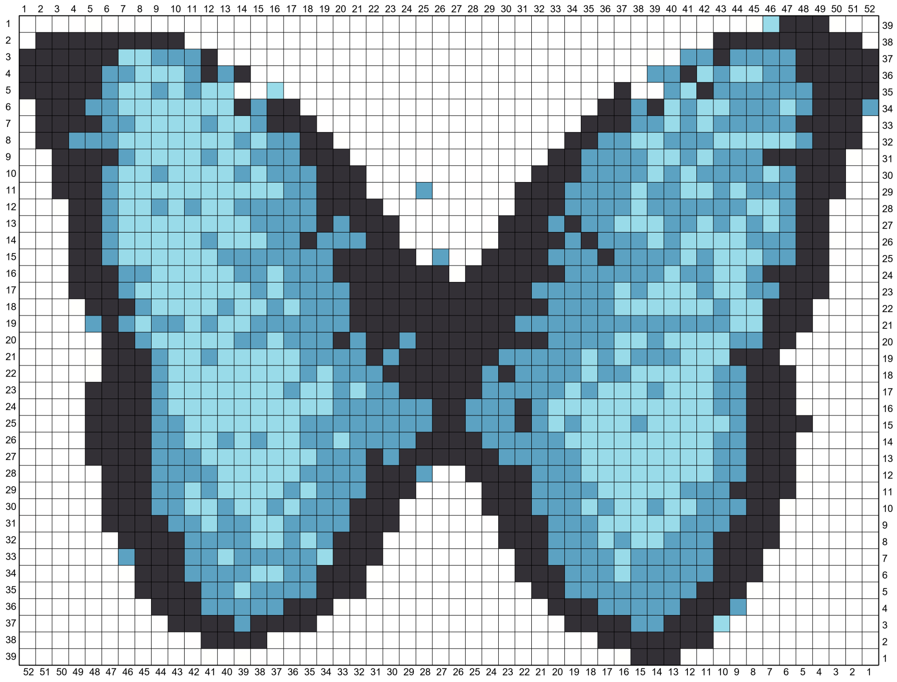
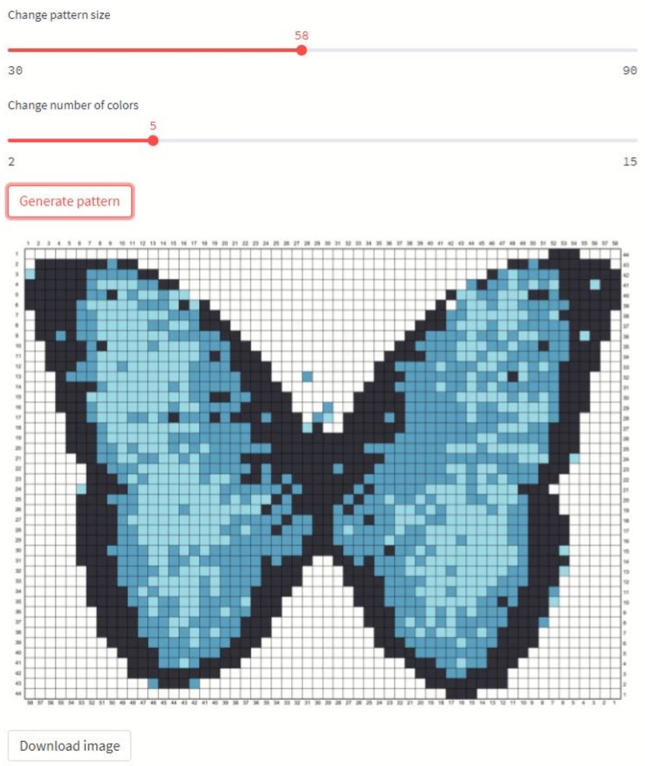


Fig. 4. C2C pattern example.

4.4 Application interface

We have two sliders in web application, where user upload the image. First slider corresponds to pattern side, second to the number of colors used. At the end of the page is a *Download pattern* button, which downloads a *.pdf* file with created pattern (see Fig. 5).



Change pattern size

58

30 90

Change number of colors

5

2 15

Fig. 5. Application interface.

5 Conclusion

This project was a great success for me personally for many reasons. Firstly, I created an application which I needed and will frequently use it in the future. Secondly, did not work with images before so this project taught me a few things about image processing. Third, I got to know a great tool for creating a quick web application.

The goal of this project was to create an application for generating C2C patterns which is realized and can be used for its purpose.

References

1. Pillow, https://pillow.readthedocs.io/. Last accessed 9 Jan 2022
2. Streamlit, https://streamlit.io/. Last accessed 9 Jan 2022
3. Analytic Vidhya, Build Web App instantly for Machine Learning using Streamlit, https://www.analyticsvidhya.com/blog/2021/06/build-web-app-instantly-for-machine-learning-using-streamlit/. Last accessed 9 Jan 2022