Online Artistic Style Transferring Service Using Fast Neural Style Implemented in Torch

Yasuo Kuhara

Tokyo Polytechnic Univ., 2-9-5 Honcho, Nakano, Tokyo 164-8678, Japan

ABSTRACT

This paper reports the development of a website for an artistic style transfer service using deep learning neural networks. This website is built using HTML and a PHP script that calls the Torch program on the web server in response to clients' requests on the Internet. Clients can select an artistic style to transfer onto their own photos. This paper contains sample source code and simple instructions for implementing relevant software to allow programmers to easily build similar artistic style transferring services on the web.

1. BACKGROUND

In recent years, I have developed a series of artworks that use artificial intelligence, and deep learning in particular. In 2017, I developed a technology to transform images into an artistic style using chainer-gogh [1] based on a neural algorithm that imparts artistic style developed by Gatys et al. [2]. This method yielded interesting results, but problems arose regarding computational speed. One transfer requires over one day of processing, even with a GPU. Thus, such technology would be impossible to apply to interactive art. In 2018, I used the chainer-fast-neuralstyle application developed by Tomoto [3], which is the Chainer implementation of Johnson's [4] perceptual losses for real-time style transfer. This method can stylize images hundreds of times faster than the Gatys method. I developed an interactive art called AI artistic painting mirror using Johnson's method [5].

In 2020, I aimed to build an artistic style transferring application as a free web service that would allow clients to easily transfer their own photos through a simple interface. In this project, I used the fast-neural-style [6] developed by Torch instead of Chainer, the development of which has finished.

```
<form action="up.php" method="post">
<h1>1. Style select</h1>
    <input type="radio" name="model" value="1">starry night
    <input type="radio" name="model" value="2">la muse
    <input type="radio" name="model" value="3">composition
    <input type="radio" name="model" value="4">wave
<h1>2. Image select</h1>
    Upload your image file.(jpeg,jpg,png)
    <input type="file" name="up_file">
    <input type="submit" value="Send">
</form>
```

Fig. 1 Outline of html script

```
<?php
$m = $_POST['model'];
$in_file = $_FILES['up_file']['tmp_name'];
$out_file = "outfile.jpg";
if($m == 1) $model =  "starry_night.t7";
if($m == 2) $model =  "la_muse.t7";
if($m == 3) $model =  "composition.t7";
if($m == 4) $model =  "the_wave.t7";
$cmd="sh exe_th.sh {$in_file} {$out_file} {$model} ";
exec($cmd);
?>
```

Fig. 2 Outline of php script (up.php)

2. SYSTEM

2.1 Web Server

I built a web server on a virtual private server/kernel-based virtual machine (VPS) system, which provides a highly flexible operating system at a low cost. The specifications of the hardware are 4 giga bytes memory, 20 giga bytes storage, 4 Core CPU, and no GPU.

The operating system is CentOS 7.9, which is a popular distribution of Linux. The http server is Apache 2.4.6, which is the most commonly used cross-platform web server software. Both are free and open-source software that can be easily and freely implemented.

2.2 HTML Script

The first page is designed for client input using HTML. In a form tag, clients select their favorite style from four models: Starry Night by Gogh, La Muse by Picasso, Composition by Kandinsky, and Wave by Hokusai. Subsequently, they select an image file to be transferred. The selected style and uploaded file are passed to a PHP script (up.php) specified in an action attribute by a post method. The sample code is shown in Figure 1.

2.3 PHP Script

The second page receives the client's uploaded file and executes a shell script (exe_th.sh) using PHP version 5.4.16, which is preinstalled in CentOS. First, PHP obtains the model selected by the client and an image file to be transferred from the previous HTML page. Next, it executes a shell script calling the Torch program of fast neural transfer using the execution function. The sample code is shown in Figure 2.

2.4 Torch Script

The shell script receives three arguments and executes

```
#!/usr/bin/bash
th fast_neural_style.lua -input_image $1 -output_image $2 -model $3
```

Fig. 3 Outline of shell script (exe_th.sh)

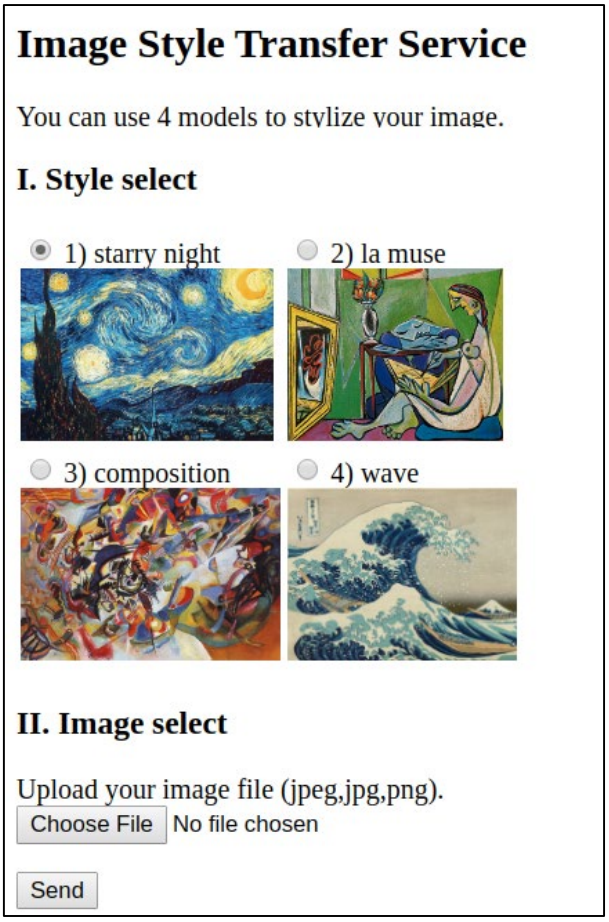

Fig. 4 User input page

a Torch program in the neural fast style written in the Lua language (fast_neural_style.lua). The sample code is shown in Figure 3.

Torch is a scientific computing framework with wide support for machine learning algorithms that is easy to use and efficient. Torch and several packages must be installed; pretrained style transfer models can be downloaded from GitHub [6]. In addition, new models can be trained by executing the Torch script. Four pretrained models were used on my website, as mentioned above.

3. EVALUATION

The website can be freely used at the linked address [7]. On the first page, clients can use four models to stylize their image files (see Figure 4); the valid file formats are JPG and PNG. The maximum upload file size is 10 megabytes. The uploaded file is reduced in size, and its longest side is 480 pixels long. Adding more memory to the server increases the resolution of the transferred image files. Clients can stylize their favorite image files in just one click. The time of the transfer was approximately 30 to 60 seconds. If a GPU is provided to the server, the time is shortened to within one second. The transferred file is shown on the next web page and can be saved in the client's local storage (see Figure 5).

Even under the limited hardware environment of the

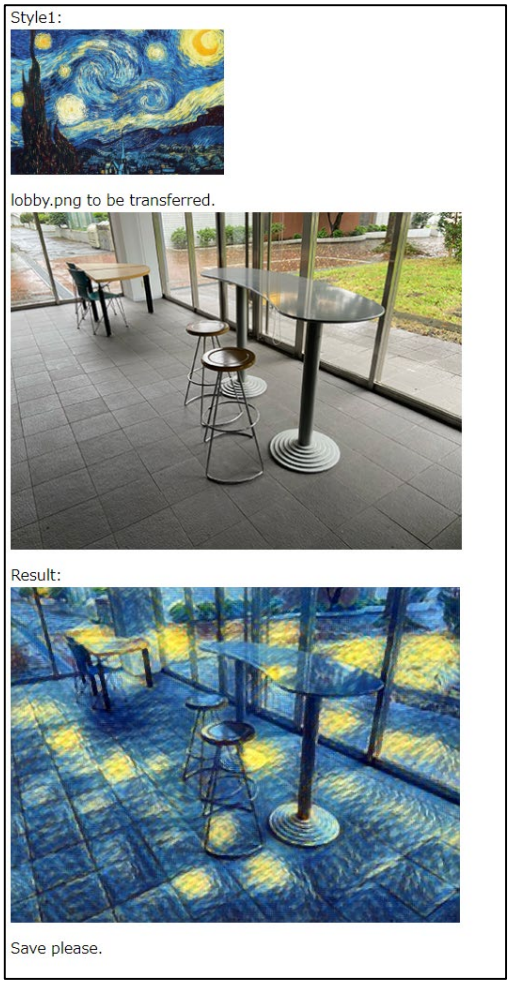

Fig. 5 Result output page

server, the results are satisfactory. In the near future, I will provide a GPU workstation as a web server to build a style transferring service to train various models for style and launch a new service for the style transfer of movies.

4. REFERENCES

[1] E. Matsumoto, "chainer-gogh", github.com/mattya/chainer-gogh (2017)
[2] L. A. Gatys et al, "A Neural Algorithm of Artistic Style", arXiv:1508.06576, Journal of Vision, Vol. 16 (2016)
[3] Y. Tomoto, "Chainer Implementation of Perceptual Losses for Real-Time Style Transfer and Super-Resolution", github.com/yusuketomoto/chainer-fast-neuralstyle (2017)
[4] J. Johnson et al, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", arXiv:1603.08155, Lecture Notes in Computer Science: pp. 694-711 (2016)
[5] Y. Kuhara, "Artificial Intelligence Artistic Painting Mirror as Interactive Art Using Deep Neural Networks," Proc. of 1st International Symposium on Color Science and Art, pp. 23-24 (2018)
[6] J. Johnson, " fast-neural-style", github.com/jcjohnson/fast-neural-style (2017)
[7] w3.kuhalabo.net/style_trans/style_trans.html (2021)