

# Lightweight Image Viewer Using Unix's Bash

github: <https://github.com/hafidhrendyanto/cliimageviewer>

13517061 Hafidh Rendyanto  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail: hafidh.rendyanto@gmail.com

**Abstract**—In the paper, we developed and tested a simple python program that can turn almost any image into an ASCII art and display it directly onto Unix's bash or terminal.

**Keywords**—component; image viewer, terminal, bash

## I. INTRODUCTION

Today, we take the ability to parse and view an image file for granted. Most of today's computers, even mobile phones can open and view any images that can be found on the internet easily. But, in today's world there's still a few cases where we don't have such as in embedded systems or remote servers that we control over ssh.

To solve that problem, in this paper we introduce a lightweight python program that can turn almost any image file into an ASCII art and display it directly onto Unix's bash or terminal.

Another usage of this program is to automatically turn any image into an ASCII art which is beloved by a niece fanbase.

This paper is inspired by the works of Andy Sloane that create a 3D engine that displays a rotating 3D model directly onto Unix's bash [1].

## II. APPROACH

### A. Prepare the Image

The first process in making the image into an ASCII art is to prepare the image so it can be displayed properly using ASCII characters and fit into Unix's bash neatly.

The first step in the preparation stage is to resize the image so that it can fit into our terminal, we lower the size of our image, setting the maximum size to be 300 pixel wide and 170 pixel tall. We try to keep the same proportion.

We are only going to use 12 ascii characters to represent each pixel which is '.,-~:;=!\*#\$@' from the lowest pixel value to the highest. As such, we need to turn our image into a grayscale image ranging from 0 to 11 in value.

### B. Preparing the Terminal

In order to have the highest resolution that we can display into our terminal, we need to reduce our terminal font size as small as possible, this is done by using few os calls.

### C. Turning the Image into ASCII Art

As we already have a grayscale image with 12 values in each pixel, turning the image into an ASCII art is easy. We just need to map each pixel value into the corresponding ASCII character that represents them.

## III. EXAMPLES

Here are a few examples of using this program to open some image file with its real representation on the side for comparison.

### 1. lenna



image iii.1.1: original image of lenna.png

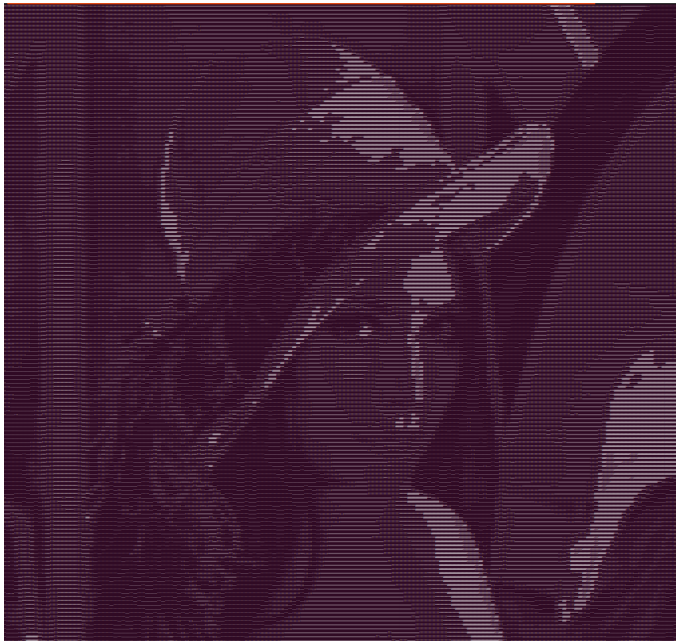


image iii.1.2: ascii art of lenna.png



image iii.2.2: ascii art of cameraman.png

2. cameraman



image iii.2.1: original image of cameraman.png

3. cats



image iii.3.1: original image of cats.png

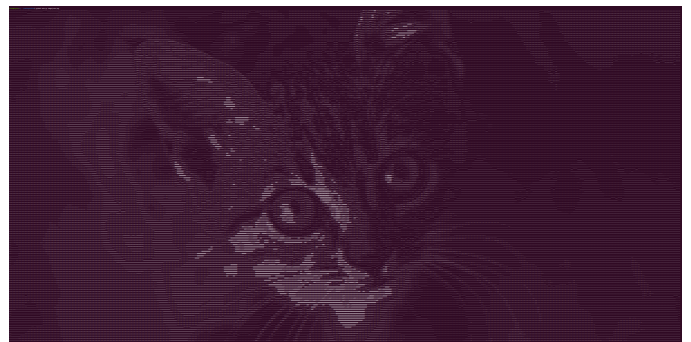


image iii.3.2: ascii art of cats.png

#### 4. blackpanther



image iii.4.1: original image of blackpanther.png

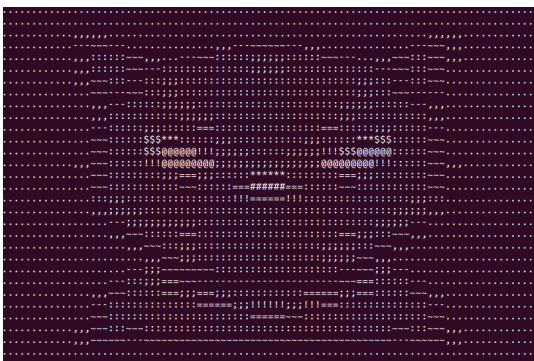


image iii.4.2: ascii art of blackpanther.png

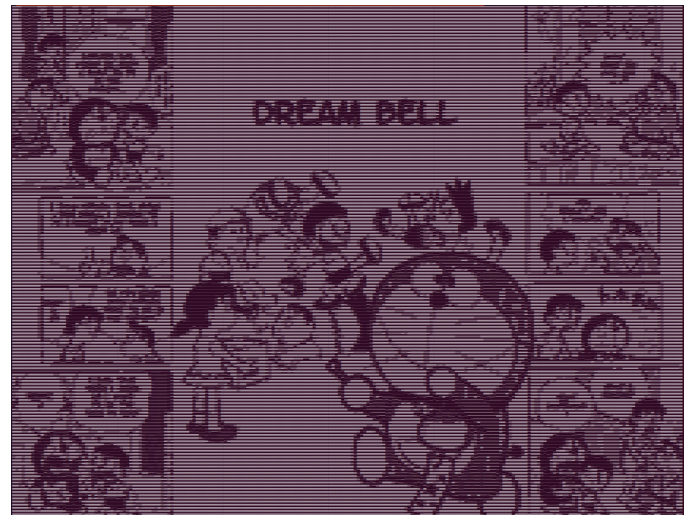


image iii.5.2: ascii art of doraemon.png

#### IV. CONCLUSION

As seen on examples, the generated ascii art properly resembles the original image in each picture although the resolution displayed on the ascii art is quite low. It is expected however, as the resource we can use in the ascii art is very limited.

The ascii art might look like a low resolution grayscale version of the original image, but we can assure you that it is made up of thousands of ascii characters, each representing a pixel.

#### REFERENCES

- [1] <https://www.a1k0n.net/2011/07/20/donut-math.html>. accessed 25 may 2021.

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 25 Mei 2021

13517061 Hafidh Rendyanto

#### 5. doraemon

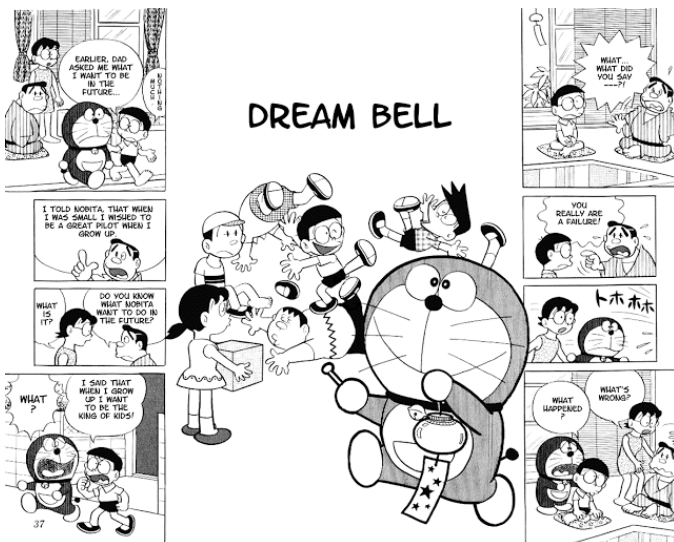


image iii.5.1: original image of doraemon.png