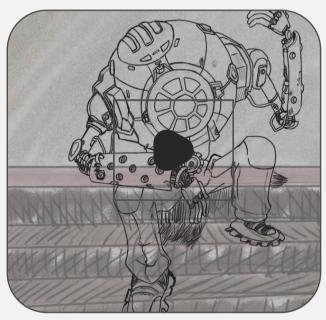


**Drawing: Old Sage** 

A drawing that I am proud of.



## **Drawing Time-Elapse**

Time-Elapse of a mecha drawing.

Tags: #Drawing #Digital

# **Projects**

```
import subprocess
import time
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract'
print("Topup Using OCR")
print("Please draw a box around the numbers you want to topup using the mouse right click button")
You.ishoursage| subhor(Wou
class Resizable|ransparentBox(tk.Tk):
    def __init__(self):
        super().__init__()
    # Configure the window
    self.overrideredirect(True)
    self.attributes('-alpha', 0.5) # Set transparency (0.8 to 1.0)
    self.attributes('-alpha', 0.5) # Set initial size and position
    # Create a label as the transparent box
    self.label = tk.label(self, text="", bg='black')
    self.label.pack(fill='both', expand='Irue)
    # Bind mouse events for dragging and resizing
    self.label.bind('sBl-Motion', self.atart_drag)
    self.label.bind('sBl-Motion', self.atart_resize)
    relf.label.bind('sBl-Motion', self.atart_resize)
    relf.label.bind('sBl-Motion', self.atart_resize)
    relf.label.pack(fill='Irue') relf.expain()
```

#### **Topup from image**

This project uses OCR (Optical Character Recognition) to read and extract numeric information from an image, enabling users to top up their accounts automatically. Built with Tkinter, PyAutoGUI, and PyTesseract, it simplifies dialing card numbers directly from the script. If connected to your device via USB with debugging enabled, it uses ADB to handle automation.

Tags: #Script #Python #PyTesseract(OCR) #PyAutoGUI #ADB

```
def jpg_to_ico(input_path, output_path):
    # Open the JPG image
    image = Image.open(input_path)

# Convert the image to RGBA (if not already in RGBA mode)
if image.mode != 'RGBA':
    image = image.convert('RGBA')

# Resize the image to 256x256 (required size for ICO format)
image = image.resize((256, 256), Image.LANCZOS)

# Save the image as ICO format
image.save(output_path, format='ICO', quality=5) # You can adjust

# Usage
input_image_path = "C:\\Users\\hp\\Desktop\\...\\just.jpg" #path to to the image output_ico_path = "C:\\Users\\hp\\Desktop\\...\\just.jpg" #Path to save the image in the
```

### Image-to-icon

This project converts image files into icons. It was inspired by my need to organize movie images into icons. When online services imposed limitations on free conversions, I created this Python-based solution using the PIL library to automate the process.

Tags: #Script #Python #PIL





Telegram



Email



Github

# Sitemap

Profile

Projects

Blogs

Contacts