|  |
| --- |
| from Colorizer import \*  import tkinter as tk  from tkinter import filedialog  class ColorizerGUI:      def \_\_init\_\_(self):          self.colorizer = Colorizer(use\_cuda = False, width = 600, height = 600)            # create root window          self.root = tk.Tk()          self.root.title("Colorization")          # create label and button for selecting input file          self.input\_label = tk.Label(self.root, text="Input File:")          self.input\_label.pack(side=tk.LEFT, padx=10, pady=10)          self.input\_entry = tk.Entry(self.root, width=40)          self.input\_entry.pack(side=tk.LEFT, padx=5, pady=10)          self.browse\_button = tk.Button(self.root, text="Browse", command=self.select\_file)          self.browse\_button.pack(side=tk.LEFT, padx=5, pady=10)          # create button for processing input file          self.process\_button = tk.Button(self.root, text="Process", command=self.process\_file)          self.process\_button.pack(side=tk.LEFT, padx=5, pady=10)          # run main loop          self.root.mainloop()      def select\_file(self):          # open file dialog to select input file          filename = filedialog.askopenfilename(initialdir="./", title="Select file", filetypes=[("Video files", "\*.mp4"), ("Image files", "\*.jpg;\*.jpeg;\*.png")])          self.input\_entry.delete(0, tk.END)          self.input\_entry.insert(0, filename)      def process\_file(self):          # process selected input file          input\_file = self.input\_entry.get()          if input\_file.endswith(".mp4"):              self.colorizer.processVideo(input\_file)          elif input\_file.endswith(".jpg") or input\_file.endswith(".jpeg") or input\_file.endswith(".png"):              self.colorizer.processImage(input\_file)          else:              print("Invalid file format")  gui = ColorizerGUI() |

|  |
| --- |
| import tkinter as tk  from tkinter import filedialog  from Colorizer import \*  import os  class ColorizerUI:      def \_\_init\_\_(self):          self.root = tk.Tk()          self.root.title("Colorization")          # Create a label for the input file selection          self.input\_file\_label = tk.Label(self.root, text="Input file:")          self.input\_file\_label.grid(row=0, column=0, padx=5, pady=5)          # Create a text box for the input file path          self.input\_file\_textbox = tk.Entry(self.root)          self.input\_file\_textbox.grid(row=0, column=1, padx=5, pady=5)          # Create a button to select the input file          self.input\_file\_button = tk.Button(self.root, text="Browse", command=self.select\_input\_file)          self.input\_file\_button.grid(row=0, column=2, padx=5, pady=5)          # Create a button to start the colorization          self.colorize\_button = tk.Button(self.root, text="Colorize", command=self.colorize)          self.colorize\_button.grid(row=1, column=1, padx=5, pady=5)      def select\_input\_file(self):          # Open a file selection dialog box and update the input file text box          file\_path = filedialog.askopenfilename()          file\_name = os.path.basename(file\_path)          self.input\_file\_textbox.delete(0, tk.END)          self.input\_file\_textbox.insert(0, file\_path)      def colorize(self):          # Get the input file path and run the colorization process          input\_file\_name = self.input\_file\_textbox.get()          colorizer = Colorizer(use\_cuda=False, width=600, height=600)          colorizer.processVideo(input\_file\_name)      def run(self):          self.root.mainloop()  if \_\_name\_\_ == "\_\_main\_\_":      colorizer\_ui = ColorizerUI()      colorizer\_ui.run() |

|  |
| --- |
| import tkinter as tk  from tkinter import filedialog  from Colorizer import \*  import os  class ColorizerUI:      def \_\_init\_\_(self):          self.root = tk.Tk()          self.root.title("Colorization")          self.colorizer = None          # Create a label for the input file selection          self.input\_file\_label = tk.Label(self.root, text="Input file:")          self.input\_file\_label.grid(row=0, column=0, padx=5, pady=5)          # Create a text box for the input file path          self.input\_file\_textbox = tk.Entry(self.root)          self.input\_file\_textbox.grid(row=0, column=1, padx=5, pady=5)          # Create a button to select the input file          self.input\_file\_button = tk.Button(self.root, text="Browse", command=self.select\_input\_file)          self.input\_file\_button.grid(row=0, column=2, padx=5, pady=5)          # Create a button to start the colorization          self.colorize\_button = tk.Button(self.root, text="Colorize", command=self.colorize)          self.colorize\_button.grid(row=1, column=1, padx=5, pady=5)          # Create a button to stop the colorization          self.stop\_colorization\_button = tk.Button(self.root, text="Stop Colorization", command=self.stop\_colorization)          self.stop\_colorization\_button.grid(row=1, column=2, padx=5, pady=5)          # Bind a function to the close button of the window          self.root.protocol("WM\_DELETE\_WINDOW", self.close\_window)      def select\_input\_file(self):          # Open a file selection dialog box and update the input file text box          file\_path = filedialog.askopenfilename()          self.input\_file\_path = os.path.abspath(file\_path)          file\_name = os.path.basename(file\_path)          self.input\_file\_textbox.delete(0, tk.END)          self.input\_file\_textbox.insert(0, file\_name)      def colorize(self):          # Get the input file path and run the colorization process          input\_file\_name = self.input\_file\_path          input\_file\_path = os.path.abspath(input\_file\_name)          self.colorizer = Colorizer(use\_cuda=False, width=600, height=600)          self.colorizer.processVideo(input\_file\_path)          self.stop\_colorization\_button.config(state="normal")          self.close\_colorization\_window\_button.config(state="normal")      def stop\_colorization(self):          # Stop the colorization process          if self.colorizer is not None:              self.colorizer.stopProcess()          self.stop\_colorization\_button.config(state="disabled")          self.close\_colorization\_window\_button.config(state="disabled")      def close\_window(self):          # Stop the colorization process and destroy the window when the close button is clicked          self.stop\_colorization()          self.root.destroy()      def run(self):          self.root.state('zoomed')          self.root.mainloop()  if \_\_name\_\_ == "\_\_main\_\_":      colorizer\_ui = ColorizerUI()      colorizer\_ui.run() |

|  |
| --- |
| # from Colorizer import \*  # import tkinter as tk  # root = tk.Tk()  # root.geometry("500x500")  # root.title("Colorization")  # root.mainloop()    # colorizer = Colorizer(use\_cuda = False, width = 600, height = 600)  # '''for img in glob.glob("input/i2.jpg"):  #     colorizer.processImage(img)  # '''  # # colorizer.processVideo("input/v1.mp4")  import tkinter as tk  from tkinter import \*  from tkinter import filedialog  from Colorizer import \*  import os, sys  class ColorizerUI:      def \_\_init\_\_(self):          self.root = tk.Tk()          self.root.title("Security - Colorization")          self.colorizer = None          self.root.geometry("400x300")          self.root.resizable(False, False)          # Create a header label          self.title = tk.Label(self.root, text="Upload Media", font=("Arial", 20))          self.title.pack(padx=10, pady=20)          # Create a label for the input file selection          self.input\_file\_label = tk.Label(self.root, text="Input file")          # self.input\_file\_label.grid(row=10, column=0, padx=10, pady=10, sticky="ew")          self.input\_file\_label.pack(padx=5, pady=5)          # Create a text box for the input file path          self.input\_file\_textbox = tk.Entry(self.root)          # self.input\_file\_textbox.grid(row=10, column=1, padx=5, pady=5)          self.input\_file\_textbox.pack(padx=5, pady=0)          self.button\_frame = tk.Frame(self.root)          # Create a button to select the input file          self.input\_file\_button = tk.Button(self.button\_frame, text="Browse", command=self.select\_input\_file)          self.input\_file\_button.grid(row=0, column=0, padx=5, pady=5)          # self.input\_file\_button.pack(pady=15)          # Create a button to start the colorization          self.colorize\_button = tk.Button(self.button\_frame, text="Colorize", command=self.colorize)          self.colorize\_button.grid(row=0, column=1, padx=5, pady=5)          # self.colorize\_button.pack()          self.button\_frame.pack(pady=15)      def select\_input\_file(self):          # Open a file selection dialog box and update the input file text box          file\_path = filedialog.askopenfilename()          self.input\_file\_path = os.path.abspath(file\_path)          file\_name = os.path.basename(file\_path)          self.input\_file\_textbox.delete(0, tk.END)          self.input\_file\_textbox.insert(0, file\_name)      def colorize(self):          # Get the input file path and run the colorization process          input\_file\_name = self.input\_file\_path          input\_file\_path = os.path.abspath(input\_file\_name)          self.colorizer = Colorizer(use\_cuda=False, width=600, height=600)          self.colorizer.processVideo(input\_file\_path)          # self.stop\_colorization\_button.config(state="normal")          # self.close\_colorization\_window\_button.config(state="normal")      # def stop\_colorization(self):      #     # Stop the colorization process      #     if self.colorizer is not None:      #         self.colorizer.stopProcess()      #     self.stop\_colorization\_button.config(state="disabled")      #     self.close\_colorization\_window\_button.config(state="disabled")      # def close\_window(self):      #     # Stop the colorization process and destroy the window when the close button is clicked      #     self.stop\_colorization()      #     self.root.destroy()      #     sys.exit(0)      # def run(self):      #     self.root.state('zoomed')      #     self.root.mainloop()  if \_\_name\_\_ == "\_\_main\_\_":      colorizer\_ui = ColorizerUI()      colorizer\_ui.root.mainloop()      # colorizer\_ui.run() |