You can replace <PROXY\_IP> and <PROXY\_PORT> with the actual values of your proxy server.

The code uses the requests library to send requests to the website and the BeautifulSoup library to parse the HTML.

The images are saved in two different folders based on their alt attribute.

The code is checking the alt attribute of the images to determine whether the image is of a person wearing clothes or a fabric. You may need to modify the conditions based on the actual website content.

The output of this program will be two folders: person\_wearing\_clothes and fabrics. Each folder will contain the images of either people wearing certain clothes or fabrics, respectively, that were scraped from the website. The images will be saved as individual files with their original file names.

The two folders, person\_wearing\_clothes and fabrics, will be stored in the current working directory where the Python script is executed.

The code uses the following libraries:

Requests: It is used to send HTTP requests to the website and retrieve the HTML content of the page.

BeautifulSoup: It is used to parse the HTML content and extract information from it. In this case, we are using it to extract the image URLs from the HTML.

Urllib: It is used to download the images from the URLs and save them to the local system.

os library: The os library is used to create directories and check if the directories already exist. The os.makedirs() method is used to create the two directories, person\_wearing\_clothes and fabrics, to store the images. The os.path.exists() method is used to check if the directories already exist, and if they do, the script will not attempt to create them again.

requests.get() method: The requests.get() method is used to retrieve the HTML content of the website. The method takes the URL of the website as a parameter and returns the HTML content in the form of a requests.Response object.

BeautifulSoup object: The BeautifulSoup object is created from the HTML content of the webpage. The beautifulsoup4 package provides the BeautifulSoup class, which is used to parse HTML and XML documents. In this case, we pass the HTML content to the BeautifulSoup constructor, along with the parser library, to parse the HTML content and create the BeautifulSoup object.

soup.find\_all() method: The soup.find\_all() method is used to search the HTML document for all elements with a specified tag. In this case, we are searching for all img tags, which contain the URLs of the images on the website. The method returns a list of Tag objects, each representing an img tag in the HTML document.

requests.get() method: The requests.get() method is used to retrieve the images from the URLs. For each img tag, we extract the URL of the image, and then use the requests.get() method to retrieve the binary content of the image.

open() method: The open() method is used to create a file and write the binary content of the image to the file. The method takes two parameters: the name of the file and the mode, which is set to wb to indicate that the file should be opened in write binary mode.

with statement: The with statement is used to ensure that the file is properly closed after the binary content of the image has been written to the file. The with statement creates a context in which the file is automatically closed after the code within the with block has been executed.

os.path.join() method: The os.path.join() method is used to construct the path to the image file, based on the name of the directory and the name of the image file. The method takes two parameters: the path to the directory and the name of the file, and returns the full path to the file

The code works as follows:

It takes the URL of the website as input and sends an HTTP GET request to the website using the requests library.

The HTML content of the page is received as a response and is then parsed using BeautifulSoup.

The parsed HTML is searched for image tags and the URLs of the images are extracted.

The URLs are then passed to the urllib library to download the images.

The images are saved to two separate folders, one for images of people dressed in certain clothes and another for images of fabrics, based on the conditions specified in the code.

In summary, the code uses the requests and BeautifulSoup libraries to extract image URLs from a website, and the urllib library to download and save the images. The images are then divided into two separate folders based on their content.

**Converting binary images dataset to .jpg**

In this code, we first import the os and Image modules from the Python Imaging Library (PIL).

The convert\_to\_jpg function takes the directory where the scraped images are stored as an argument. The function then loops through all the files in the directory, checking if they have a .png, .jpeg, or .gif extension.

For each of these files, the function uses the Image.open method from PIL to open the file and convert it to a RGB image. The converted image is then saved in JPEG format with a quality of 90. Finally, the original file is removed.

The convert\_to\_jpg function is called with the directory path of the scraped images folder as an argument. This will convert all the images in the folder to .jpg format.

The cv2 library in folder "Part2" is used for image processing, including reading and writing images, masking, and adding weighted images. The numpy library is used for creating and manipulating arrays, such as the blue background image.The code preprocesses an image of a cloth and its corresponding cloth mask to generate a new image with the blue background. The input to the function "preprocess" are the original cloth image and its cloth mask, which is a black and white image indicating the foreground (cloth) and the background.

The function takes these inputs and combines them to produce an output image with the original cloth and a blue background. This process is achieved through image processing techniques such as bitwise operations and color channels manipulations. The output image is saved or returned as a result of the function.

This code can be applied to all images in a dataset to generate new images with blue backgrounds, and it serves as a useful tool for image processing and computer vision applications.

To summarize my work i want to tell that the BeautifulSoup and Requests libraries are very important for web scraping because they provide the tools necessary for parsing and extracting information from web pages.

BeautifulSoup is a Python library for pulling data out of HTML and XML files. It allows you to parse HTML and XML pages and extract information such as text, links, images, and other elements. With BeautifulSoup, you can easily navigate and search a page's HTML structure to find the information you need.

Requests is a Python library for sending HTTP requests. It allows you to make HTTP requests, such as GET and POST requests, to web servers and receive their responses. This library makes it easy to send requests to a web server, retrieve its HTML source code, and parse the information using BeautifulSoup.

Together, BeautifulSoup and Requests provide a powerful toolset for web scraping, as they allow you to programmatically access and extract information from websites. These libraries are particularly useful for tasks like downloading images, collecting data from online sources, and automating online tasks.