```
import pandas as pd
from PIL import Image
import requests
from io import BytesIO
import os

filtered_df = pd.read_csv('data/topwear.csv')

#filtered_df = filtered_df.head(10000)

red_values = []
green_values = []
blue_values = []

def process_image(link, index):
    try:
        # requesting image
        response = requests.get(link, timeout=10)
        response.raise_for_status()
        img = Image.open(BytesIO(response.content))

        # cropping
        width, height = img.size
        crop_box = (
            width * 0.375,
            height * 0.375,
            width * 0.625,
            height * 0.625
        )
        cropped_img = img.crop(crop_box)

        # RGB from image, source code https://stackoverflow.com/questions/13137817/how-to-
download-image-using-requests

        cropped_img = cropped_img.convert("RGB")
        pixels = list(cropped_img.getdata())
        num_pixels = len(pixels)


        avg_red = sum(pixel[0] for pixel in pixels) / num_pixels
        avg_green = sum(pixel[1] for pixel in pixels) / num_pixels
        avg_blue = sum(pixel[2] for pixel in pixels) / num_pixels

        print(f"Processed image {index + 1}/{len(filtered_df)}: {link}")
        return avg_red, avg_green, avg_blue
    except Exception as e:
        print(f"Error processing image {index + 1}/{len(filtered_df)} ({link}): {e}")
        return None, None, None

for idx, link in enumerate(filtered_df['link']):
    r, g, b = process_image(link, idx)
    red_values.append(r)
    green_values.append(g)
    blue_values.append(b)

filtered_df['R'] = red_values
filtered_df['G'] = green_values
filtered_df['B'] = blue_values

cols = list(filtered_df.columns)
id_index = cols.index('id')
cols = cols[:id_index + 1] + ['R', 'G', 'B'] + cols[id_index + 1:-3]
filtered_df = filtered_df[cols]

output_path = 'data/rgb.csv'
os.makedirs(os.path.dirname(output_path), exist_ok=True)
filtered_df.to_csv(output_path, index=False)
```

```
print(f"RGB values extracted and saved to {output_path}")
```