Gradient Obfuscation

Name	Roll
Umair Sabir	200992

Introduction:

My obfuscation script is a Python program that combines three image processing techniques: denoising, enhancement, and style transfer, to obfuscate an input image. The script uses the OpenCV library for image processing tasks and the BM3D library for denoising. The purpose of this obfuscation is to create a visually distinct image that maintains a certain level of quality while making it more difficult to discern the original content.

Technical Details:

Libraries and Dependencies:

- OpenCV library (cv2): Used for image processing tasks, such as reading and writing images, color space conversion, and applying the style transfer model.
- BM3D library: Used for denoising the input image.
- urllib.request and os modules: Used for downloading the style transfer models and checking if they already exist on the local machine.

Functions:

`download_model(url, model_path)` `:` Downloads the style transfer model from the given URL and saves it to the specified model path if it doesn't already exist on the local machine.

`apply_denoising(image, denoising_strength=1.0)` `:` Applies the BM3D denoising algorithm to the input image with the specified denoising strength.

`apply_style_transfer(image, style='the_wave')` `:` Applies the chosen style transfer model to the input image. The user can choose from three styles: 'the_wave', 'starry_night', or 'la_muse'.

`enhance_image(image)` `:` Enhances the input image by equalizing the histogram of the Y channel in the YCrCb color space, improving the contrast of the image.

`obfuscate_image(image_path, style='the_wave', denoising_strength=1.0)` `:` Combines the denoising, enhancement, and style transfer functions to obfuscate the input image.

Workflow:

- The script reads the input image using the OpenCV `cv2.imread()` function.
- The input image is denoised using the `apply_denoising()` function, which utilizes the BM3D denoising algorithm.
- The denoised image is enhanced using the `enhance_image()` function, which equalizes the histogram of the Y channel in the YCrCb color space.
- The enhanced image is passed through the chosen style transfer model using the `apply_style_transfer()` function. The style transfer models are pre-trained neural networks that have been trained on famous artistic styles.
- The obfuscated image is saved to the specified output path using the OpenCV `cv2.imwrite()` function.

Usage:

- The user needs to provide the input image path, output image path, desired style, and denoising strength.
- The script reads the input image, applies the denoising, enhancement, and style transfer techniques, and saves the obfuscated image to the specified output path.

Uniqueness:

This script is unique and potentially better than some other obfuscation methods due to the combination of three image processing techniques: denoising, enhancement, and style transfer. This combination allows the script to produce obfuscated images that are not only visually distinct but also maintain a certain level of quality.

1. **Denoising**: The use of the BM3D denoising algorithm helps remove noise from the input image, which can be particularly useful when working with low-quality or noisy images. By reducing noise, the script ensures that the subsequent enhancement

and style transfer steps work on a cleaner image, which can lead to better obfuscation results.

2. **Enhancement**: The histogram equalization of the Y channel in the YCrCb color space improves the contrast of the image. This step enhances the visual quality of the image, making it more visually appealing and potentially harder to reverse-engineer or recognize the original content.
3. **Style Transfer:** The neural style transfer models used in the script are pre-trained on famous artistic styles, such as 'the_wave', 'starry_night', and 'la_muse'. By applying these styles to the input image, the script creates an obfuscated image that is visually distinct and bears the characteristics of the chosen artistic style. This makes it more difficult for an observer to discern the original content of the image.

Remarks:

In summary, the uniqueness and potential advantages of this obfuscation script lie in the combination of denoising, enhancement, and style transfer techniques. By applying these techniques sequentially, the script can produce obfuscated images that are visually distinct, maintain a certain level of quality, and are potentially more challenging to reverse-engineer or recognize the original content.

References:

```
https://github.com/jcjohnson/fast-neural-style
```

https://arxiv.org/abs/1508.06576

https://docs.opencv.org/master/d6/d00/tutorial_py_root.html

https://opencv.org/

https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html

https://github.com/linhlinhle997/style-transfer/tree/master/models/eccv16