Как использовать данный код?

Сначала нам нужно установить все библиотеки, которые указаны в файле requirements.txt. Вы можете зайти в этот файл и самостоятельно установить всё то, что там указано (используя именно те версии, которые тоже там указаны!). Но есть более простой способ это установить. Можно воспользоваться программой pip (в новых версиях python она автоматически тоже устанавливается). Но если этой программы нет – её можно установить. Для этого можно просто зайти в консоль (не важно, в какой директории Вы находитесь), после чего выполнить команду:

**sudo apt-get install python3-pip**

После этого мы можем использовать программу pip следующим образом:

**pip install -r requirements.txt**

Для того, чтобы использовать написанную программу, нам необходимо находится в директории, в которой эта программа расположена.

Для этого мы открываем консоль, и используем встроенную Linux программу cd (change directory). Как мы её используем? Допустим, наш проект находится в папке new, которая находится в нашей домашней директории (home). Тогда мы используем следующую команду, чтобы перейти в директории проекта:

**cd /home/new**

или

**cd ~/new**

Теперь подробнее о самой программе. Как её использовать? Допустим, у нас есть картинка и цифровой водяной знак, и мы хотим получить картинку вместе с ЦВЗ (то есть хотим использовать модуль встраивания).

Для этого мы должны использовать следующую команду:

**python main.py --origin <path\_cover\_image> --ouput <path\_output\_image>**

Приведём конкретный пример для большего понимания. Введём следующее:

**python main.py --origin cover.jpg --ouput watermarked.jpg**

После выполнения команды мы выбираем алгоритм, который хотим использовать (DCT или DWT), после чего выбираем “embedding”.

Далее, если мы хотим получить обратно водяной знак из картинки со встроенным ЦВЗ, мы должны использовать следующую команду:

**python main.py --origin <path\_watermarked\_image> --ouput <path\_extracted\_signature>**

Снова приведём конкретный пример. Вводим, например:

**python main.py --origin watermarked.jpg --ouput signature.jpg**

Снова выбираем между DCT и DWT, после чего выбираем “extracting”.

Ещё в этой программе реализована дополнительная функция. Мы можем не только встраивать ЦВЗ и извлекать ЦВЗ – мы можем взять ту или иную картинку с ЦВЗ и попытаться получить картинку, которая уже не имеет ЦВЗ (то есть избавляемся от ЦВЗ). Для этого мы должны использовать следующий шаблон:

**python main.py --origin <path\_watermarked\_image> --ouput <path\_attacked\_image>**

Снова приведём конкретный пример:

**python main.py --origin watermarked.jpg --ouput watermarked.jpg**

После выполнения команды выбираем “Attack” (такой метод называют атакой на изображение с водяным знаком), после чего выбираем тип атаки.

Для обучения этой нейронной сети были использованы датасеты ImageNet, CIFAR-10, а также собственные изображения.

Не забывайте перемещать картинки, с которыми желаете работать, в папку new.

Теперь чуть подробнее о самой программе. В ней мы реализуем алгоритмы DCT, DWT нанесения водяных знаков. DCT расшифровывается как Discrete Wavelet Transform, а DWT – как Singular Value Decomposition.

Программа использует DWT для разбиения изображения на его частотные компоненты и применяет SVD для встраивания водяного знака в низкочастотные компоненты.

Также в программе используются такие библиотеки, как NumPy, OpenCV и PyWavelets.