МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

Кафедра Инженерной кибернетики

Курсовая работа

по дисциплине «Интеллектуальная обработка изображений»

на тему:

«Разработка алгоритма перекраски серых изображений в цветные на основе opencv/caffe»

Выполнил ст. гр. МПИ-19-2-2:

Льянов А. Р.

Принял:

Садеков Р. Н.

Москва, 2020-2021

**Постановка задачи**

В рамках индивидуальной исследовательской работы необходимо разработать алгоритм перекраски серых изображений в цветные. Разработанный алгоритм нужно реализовать на языке Python. Кроме того, необходимо использовать библиотеку opencv для работы с изображениями и модуль caffe для получения модели нейронной сети для перекраски изображения.

Отчет должен содержать результаты работы разработанного алгоритма, текстовую часть с обоснованием выбранного подхода, ссылку на «Github» хранилище с кодом программы и инструкцией по ее запуску на ОС «Windows».

**Алгоритм**

Алгоритм разрабатывается, в основном, для раскрашивания фотографий, которые были сделаны в то время, когда все фотографии были черно-белыми. В то время качество фотографий оставляло желать лучшего, поэтому на изображениях очень часто присутствуют шумы. В основном, это шумы типа соль-перец.

Для того, чтобы изображение было раскрашено как можно лучше, нужно постараться эти шумы убрать. Для удаления шумов обычно используются различные фильтры, такие как:

* Гауссовский фильтр;
* Средний фильтр;
* Медианный фильтр;
* Лаплассовский фильтр;
* Частотный фильтр;
* Консервативный фильтр;
* И др.

Для удаления шумов типа соль-перец лучше всего подходят средний, гауссовский, медианный и консервативный фильтры. При этом средний фильтр сильно размывает изображение, поэтому его использовать не будем. Гауссовский фильтр не сильно отличается от среднего, поэтому его мы тоже не будем использовать.

Из фильтров, хорошо проявляющих себя в борьбе с шумами, остаются медианный и консервативный. Они и будут использоваться в алгоритме для подготовки изображения к перекраске.

Медианный фильтр вычисляет медиану интенсивностей пикселей, которые находятся возле центрального пикселя, и заменяет значение центрального пикселя на полученную медиану.

Консервативный фильтр определяет максимальную и минимальную интенсивности в окрестности некоторого пикселя, не включая интенсивность его самого. Далее, если интенсивность центрального пикселя превышает максимальное значение, то его интенсивность заменяется максимальным значением. Если его интенсивность ниже минимального значения, то она заменяется минимальным значением.

И консервативный, и медианный фильтры хорошо устраняют шум типа соль-перец, но плохо устраняют спекл-шум. К тому же, оба фильтра неплохо сохраняют края изображения и не сильно смазывают его, особенно при маленьких размерах ядра.

Исходя из этого, чтобы сильно не смазывать изображение, было выбрано ядро размера 3х3 для медианного фильтра и ядро размера 5х5 для консервативного фильтра.

После предобработки изображения необходимо как-то предать ему цвет.

Это решено было сделать следующим образом:

* Перевести изображение в формат LAB. Здесь L отвечает за «светлоту» изображения, а A и B – за цветовой тон и насыщенность.
* Выделить параметр L. Так как наше изображение серое, то, по сути, оно состоит только из светлоты (интенсивности). Поэтому по параметру L можно каким-то образом спрогнозировать параметры A и B.
* Изменить размер изображения на [224, 224], так как нейронная сеть принимает изображения только такого размера.
* С использованием модели нейронной сети, подавая ей на вход матрицу параметров L каждого пикселя, прогнозируем параметры A и B для каждого пикселя.
* Объединяем параметры L, A и B для того, чтобы получить раскрашенное изображение.

В качестве нейронной сети для перекраски изображения была использована вторая версия уже готовой нейронной сети, которую можно найти в модуле caffe: colorization\_release\_v2.caffemodel.

Стоит отметить, что вроде как это единственный доступный обученный вариант на данный момент, который можно получить в свободном доступе.

**Программная реализация**

Программная реализация была выполнена на языке Python версии 3.8.3. Для графической оболочки был использован модуль PyQt5 – набор расширений графического фремворка Qt для языка Python. Для работы с изображениями была использована библиотека OpenCV.

Из модуля caffe была взята нейронная сеть для перекраски изображений.

Ссылка на Github: <https://github.com/alexlyanov/colorization>.

**Пример работы алгоритма**

Исходное изображение:

Изображение выглядит как текст, человек, внешний, старый

Автоматически созданное описание

Переведенное в серый изображение с измененным размером [224, 224].



Изображение после применения медианного фильтра:



Изображение после применения консервативного фильтра:



Итоговое изображение:

Изображение выглядит как текст, внешний, человек

Автоматически созданное описание

Для сравнения пример результата, который получается, если предварительно не обработать изображение с помощью медианного и консервативного фильтров (заметно, что из-за шума цвета получились не такими насыщенными, особенно зеленый):

Изображение выглядит как текст, внешний, человек

Автоматически созданное описание

**Список использованных источников**

1. Документация PyQt5. [Электронный ресурс] <https://python-scripts.com/pyqt5>
2. Документация OpenCV. [Электронный ресурс] <https://docs.opencv.org/master/d6/d00/tutorial_py_root.html>
3. Предварительная обработка изображений. [Электронный ресурс] https://ru.bmstu.wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%B2%D0%B0%D1%80%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F\_%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0\_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9
4. Гарсия Глория Буэно, Суарес Оскар Дениз. Обработка изображений с помощью OpenCV, 2016.