

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ПРИЛОЖЕНИЕ ДЛЯ ПЕРЕНОСА ГРАФИЧЕСКОГО СТИЛЯ
НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Новиковой Анастасии Валерьевны

Научный руководитель
доцент, к. ф.-м. н.

С. В. Миронов

Заведующий кафедрой
к. ф.-м. н., доцент

А. С. Иванов

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 3 |
| 1 Основное содержание работы | 4 |
| 2 Графическая библиотека Tkinter | 5 |
| 3 Разработка графического интерфейса | 6 |
| 4 Реализация фильтров обработки изображений | 7 |
| ЗАКЛЮЧЕНИЕ | 16 |

ВВЕДЕНИЕ

С каждым новым днем все больше людей стремятся запечатлеть удачный момент, памятный день, сфотографировать красивый пейзаж и поделиться сделанными снимками со своими друзьями или знакомыми. Именно по этой причине интерес к программам для обработки изображений растет в геометрической прогрессии, как и количество графических редакторов для их преобразования.

На данный момент широко известны следующие программы: GIMP, Adobe Photoshop, Microsoft Paint, Microsoft Photo Editor, Artipic, Corel PhotoPaint, Corel Painter, Pixelmator, Paint.NET, Krita, MyPaint, Pinta, Rawstudio, PhotoFiltre, KolourPaint, Tux Paint и многие другие. Каждая из этих программ имеет свои особенности и недостатки, определенные «фичи». Такое количество уже существующих редакторов показывает актуальность разработки программ со схожим функционалом и с новыми особенностями. Данная работа посвящена разработке программного пользовательского интерфейса для обработки изображений. Стоит отметить, что необходимо создать такой графический редактор, который бы в какой-то степени отличался от «конкурентов». В качестве особенности разрабатываемого редактора предполагается реализовать фильтры для стилизации изображения, основанные на применении нейронных сетей.

Целью настоящей работы является разработка программного пользовательского интерфейса для обработки изображений, в котором реализуются возможности стилизации изображения.

Для достижения данной цели были поставлены следующие задачи:

- изучить принципы работы сверточных нейронных сетей, входящих в состав технологий глубокого обучения;
- изучить возможности библиотеки TensorFlow для последующего ее использования в работе;
- изучить особенности и возможности среды разработки Keras;
- приобрести навык работы с моделью VGG-19;
- разработать алгоритм стилизации изображений с использованием нейронных сетей;
- реализовать удобный в использовании пользовательский графический интерфейс, который позволит упростить процесс применения фильтров.

1 Основное содержание работы

Бакалаврская работа включает в себя введение, три раздела, 50 рисунков, заключение, список использованных источников и два приложения. Общий объем работы составляет 88 страниц. Список источников включает в себя 25 наименований. Первый раздел посвящен краткому обзору возможностей и особенностей библиотеки Tkinter, использование которой предоставляет возможность создания приятного глазу и удобного в использовании GUI. Во втором разделе описаны и проиллюстрированы основные возможности разработанного графического программного интерфейса. Третий раздел является самым объемным и включает в себя большое количество подразделов, поскольку в нем описывается разработка алгоритма нейронного переноса стиля, реализация фильтров для стилизации изображений, основанных на применении нейронных сетей и классического набора фильтров, включающего в себя следующие эффекты: черно-белое изображение, сепия, шум, рисунок углем, негатив.

2 Графическая библиотека Tkinter

Язык программирования Python предлагает некое множество различных вариантов разработки графического пользовательского интерфейса. Если рассматривать все методы GUI, то можно утверждать, что Tkinter является одним из самых часто используемых методов. Это стандартный интерфейс Python для инструментария Tk GUI, поставляемый с Python. Tkinter является событийно-ориентированной графической библиотекой, которая позволяет создавать программы с оконным интерфейсом. Используя Python в совокупности с библиотекой Tkinter, получаем самый быстрый и простой способ создания приложений с графическим интерфейсом.

3 Разработка графического интерфейса

После компиляции программного кода пользователю становится доступна возможность взаимодействия с программным интерфейсом, разработанным при помощи библиотеки Tkinter (рис. 1).

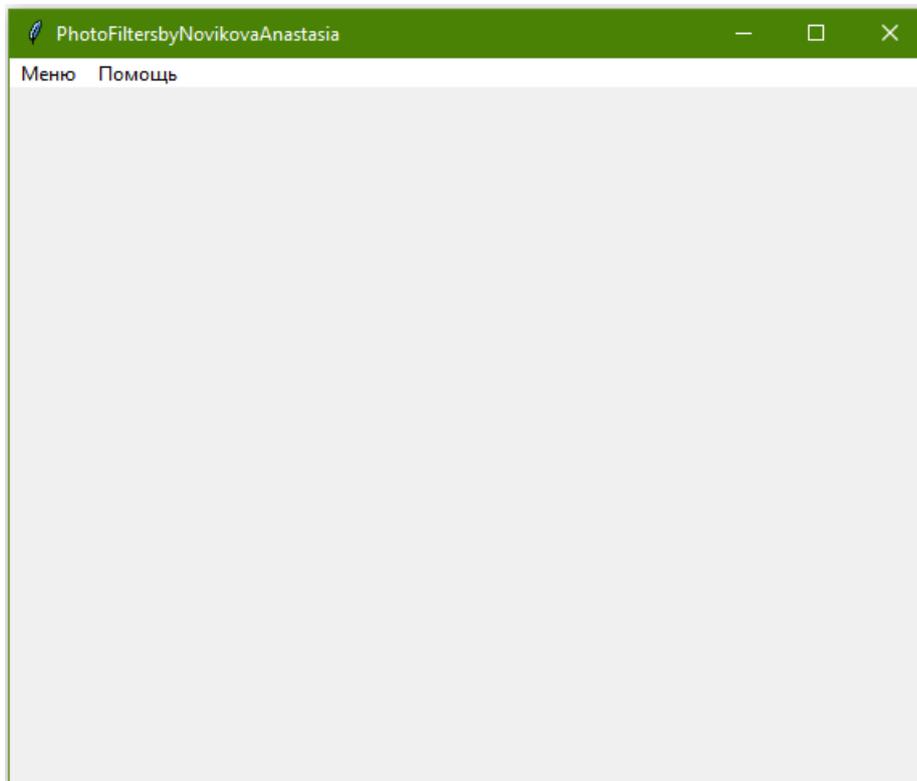


Рисунок 1 – Начало работы

В разделе «Меню» можно увидеть опции, которые предоставляет интерфейс. Чтобы перейти к ознакомлению со справочной информацией, необходимо в разделе «Помощь» выбрать пункт «Справка». Справка включает в себя описание основных функций и фильтров для обработки изображений, реализованных в данном программном интерфейсе.

4 Реализация фильтров обработки изображений

Данный раздел состоит из 6 подразделов. В первых трех подразделах внимание уделяется тому, что было использовано в ходе разработки нейронного переноса стиля.

Первый подраздел включает в себя информацию о сверточных нейронных сетях, которые являются основным инструментом для распознавания различных объектов, а также лиц на фотографиях и их классификации. Основным отличием сверточных нейронных сетей от всех других является применение специальной универсальной операции, которая называется «сверткой». Она позволяет выделять ребра, грани или контуры, то есть опорные признаки фотографий.

Второй подраздел посвящен описанию библиотек Keras и TensorFlow, которые были использованы в данной работе. TensorFlow является одной из самых лучших библиотек, которые используются для глубокого обучения. Компания «Google» разработала данную библиотеку с открытым исходным кодом для численного расчета с использованием графа потока данных. Узлы на таком графе представляют собой математические операции, а ребра этого графа представляют собой многомерные массивы данных, передаваемые между ними. Такие массивы данных называют «тензорами». Из-за своих возможностей и функционала TensorFlow заслуженно считается очень мощным инструментом для разработки. Однако, стоит отметить, что TensorFlow является довольно сложной для использования библиотекой, а потому требует достаточно длительного предварительного изучения архитектуры и всех особенностей. Для того чтобы облегчить использование TensorFlow, была разработана библиотека Keras, которую можно назвать «оберткой» для TensorFlow.

В третьем подразделе описана сетевая архитектура VGG-19, состоящая из 19 слоев. Так как нейронный перенос стиля опирается на сверточный тип заранее обученной нейронной сети (CNN), то в качестве модели сверточной нейронной сети была взята сеть VGG-19. Это нейронная сеть глубокого обучения, разработанная Visual Geometry Group для распознавания объектов на фотографиях. Она обучена на базе данных «ImageNet». «ImageNet» представляет собой массивную базу изображений. Она включает в себя более 14 миллионов изображений и более 20 тысяч категорий, что является впечатляющими числами. Модели VGG можно использовать для прогнозирования, извлечения

признаков и точной настройки. Веса загружаются автоматически при создании экземпляра модели.

Целью данной работы являлось создание такой нейросети, которая могла бы принимать на вход обработки абсолютно любую фотографию, а на выходе представляла бы изображение, которое выглядит так, будто оно является картиной, написанной рукой великих художников, чей стиль невозможно не узнать. Необходимо добиться такого результата, чтобы в нейросети успешно отделялись репрезентации контента, то есть исходного изображения, поданного на вход, и непосредственно стиля. Это позволит управлять ими независимо друг от друга. Так, мы сможем взять стиль одного изображения, а от другого взять контент. В этом и будет заключаться вся суть нейронного переноса стиля, разработка которого подробно описана в четвертом подразделе.

В данном подразделе описываются особенности моментального или же стремительного исполнения в TensorFlow, которое было использовано в работе. Помимо этого, в подразделе содержится информация о промежуточных слоях, которые играют огромную роль в процессе определения представлений. Для исходной фотографии необходимо сопоставить соответствующие представления на этих промежуточных слоях. Если мы хотим получить представления выбранного нами изображения и изображения стиля, то нам необходимо смотреть на промежуточные слои модели. Под такими слоями подразумеваются карты признаков, которые становятся все более и более упорядоченными в процессе постепенного углубления. Также был описан способ, как можно подгрузить подготовленную предобученную сеть классификации изображений.

Описания реализации функции потерь для контента и реализации функции потерь для стиля также присутствуют в четвертом подразделе. Был описан градиентный спуск и расчет градиентов, так как нам необходимо минимизировать потери при переносе стиля, описан оптимизатор Adam.

В пятом подразделе происходит демонстрация работы нейронного переноса стиля. В качестве изображений стиля были выбраны несколько картин известных художников. Для переноса стиля картин Винсента Ван Гога в качестве изображения стиля была выбрана картина «Звездная ночь», являющаяся одной из самых известных картин нидерландского художника-постимпрессиониста.

Сначала, используя возможности нашего созданного графического ин-

терфейса, открываем фотографию. Данный шаг показан на рисунке 2.

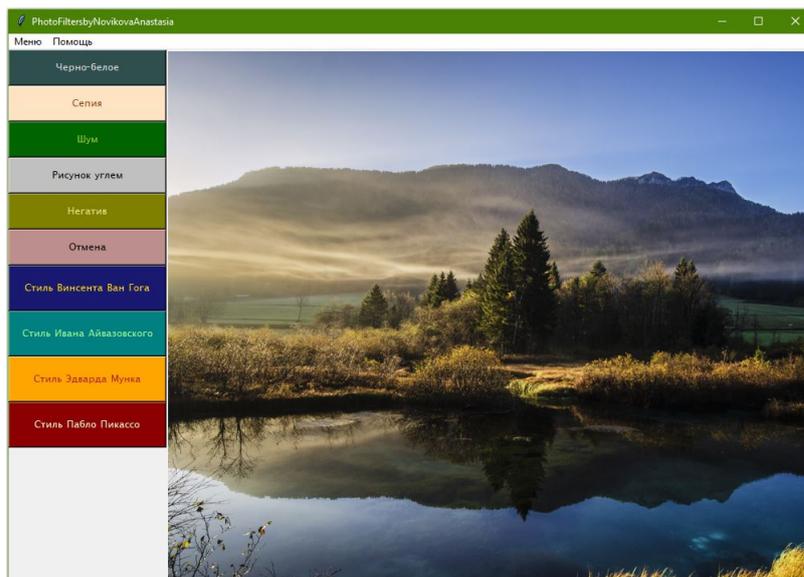


Рисунок 2 – Выбранное пользователем исходное изображение

Выбранное нами изображение отрисовалось на канвасе, то есть оно открылось. Теперь нажимаем на кнопку «Стиль Винсента Ван Гога» и наблюдаем за преобразованиями. После нажатия на кнопку происходит визуализация входных данных, а именно: изображение контента (выбранной нами фотографии) и изображение стиля (рис. 3).

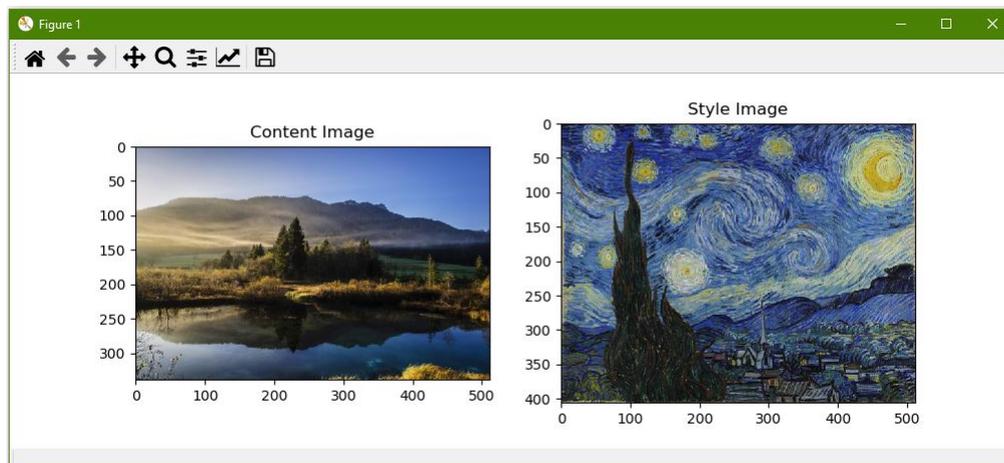


Рисунок 3 – Изображение контента (выбранной нами фотографии) и изображение стиля

После окончания выполнения нейронного переноса стиля будет визуализирован процесс выполнения итераций, полученное изображение, а также непосредственно на канвасе будет отрисован результат работы фильтра.

Так, на рисунке 4 показаны промежуточные результаты работы нейронного переноса стиля на 1, 101, 201, 301, 401, 501, 601, 701, 801 и 901 итерациях.



Рисунок 4 – Промежуточные результаты работы нейронного переноса стиля

Осуществляется отрисовка полученного результата на канвасе нашего графического интерфейса.

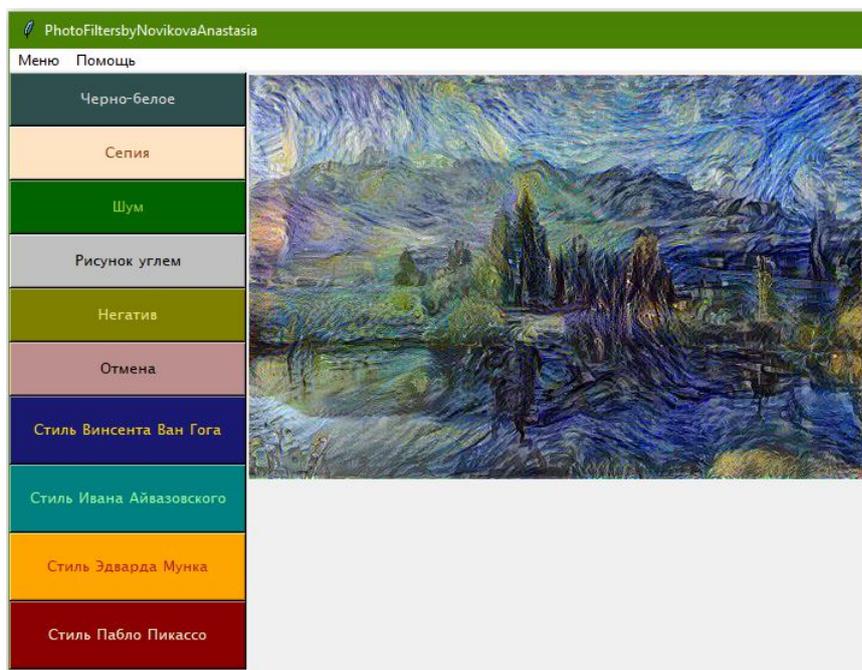


Рисунок 5 – Результат применения фильтра, выполняющего стилизацию

Аналогично осуществляется перенос стиля картин Ивана Айвазовского, Эдварда Мунка и Пабло Пикассо. На рисунках 6 и 7 можно увидеть результаты работы нейронного переноса стиля для картины Ван Гога «Звездная ночь» и для картины «Девятый вал» Айвазовского. Казалось бы, мы выбрали одно и то же изображение для его последующего редактирования, однако, результат просто поражает. Полученные изображения выглядят абсолютно по-разному, но схожи по «содержанию». Это служит прекрасной демонстрацией принципа работы нейронного переноса стиля, который был описан ранее. Полученное изображение выглядит, как то, что было подано на вход, но в то же время

совсем по-другому из-за того, что был перенесен стиль с работ разных художников, что мы, собственно, и можем наблюдать на представленных ниже изображениях.

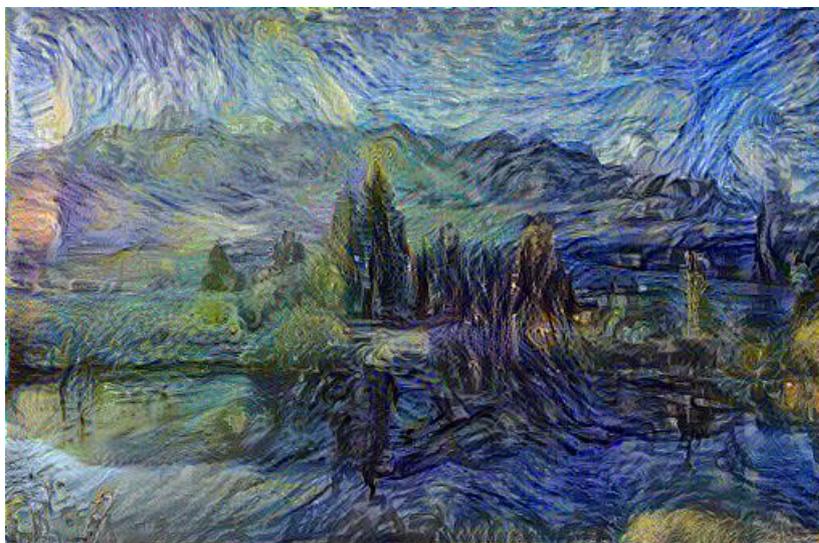


Рисунок 6 – Результат, полученный после применения эффекта «Стиль Винсента Ван Гога»

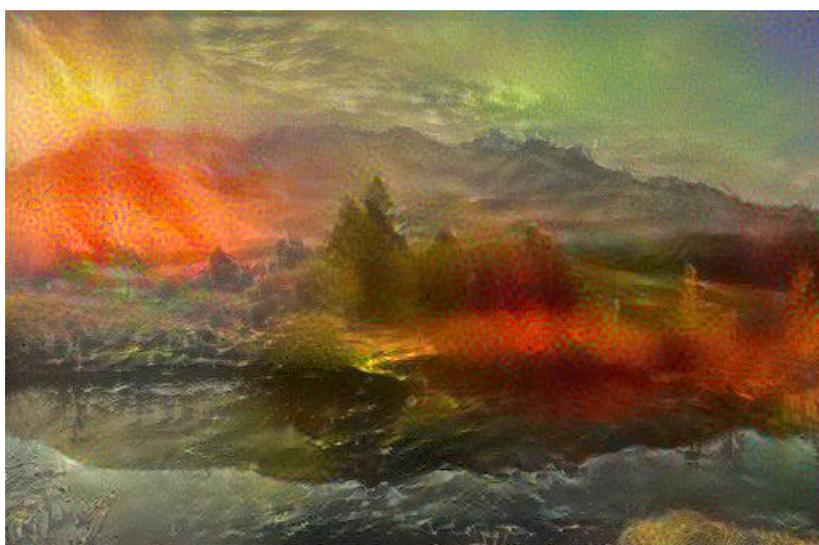


Рисунок 7 – Результат, полученный после применения эффекта «Стиль Ивана Айвазовского»

На рисунке 8 отображается результат, полученный после применения эффекта «Стиль Эдварда Мунка». В качестве изображения стиля была выбрана картина «Крик».

В качестве изображения стиля для эффекта «Стиль Пабло Пикассо» была выбрана картина «Клетка для птиц». На рисунке 9 показан результат наложения этого фильтра.

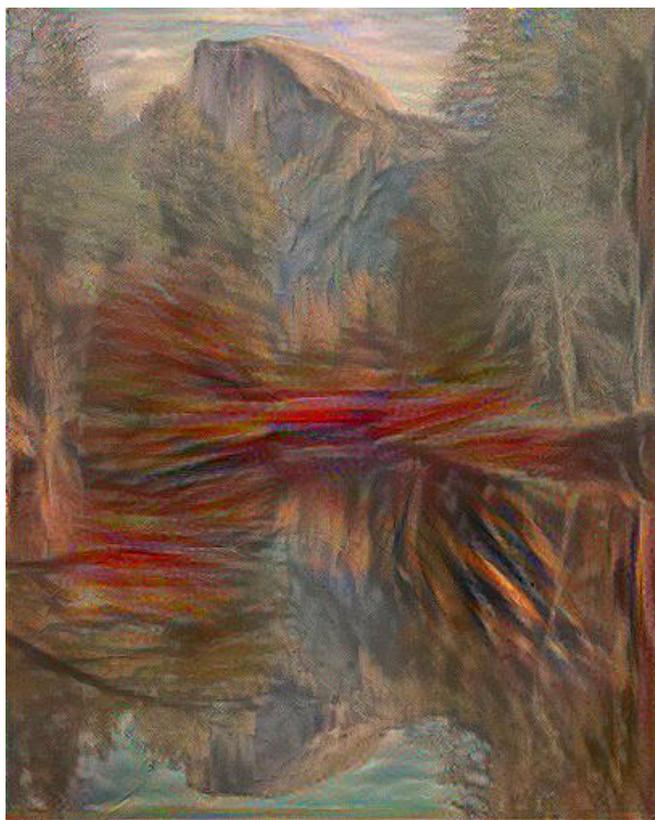


Рисунок 8 – Результат, полученный после применения эффекта «Стиль Эдварда Мунка»

В шестом подразделе описаны алгоритмы разработки классических эффектов. К ним можно отнести следующие эффекты: черно-белый фильтр (рис. 10), сепия (рис. 11), шум (рис. 12), рисунок углем (рис. 13) и негатив (рис. 14).

Пользователю необязательно ограничивать себя выбором какого-то одного фильтра. Так, он может как применять различные эффекты поочередно, так и выполнить наложение одного и того же эффекта несколько раз. Результат может быть очень интересным и необычным, все зависит от воображения пользователя.



Рисунок 9 – Обработанная фотография, сохраненная в директорию

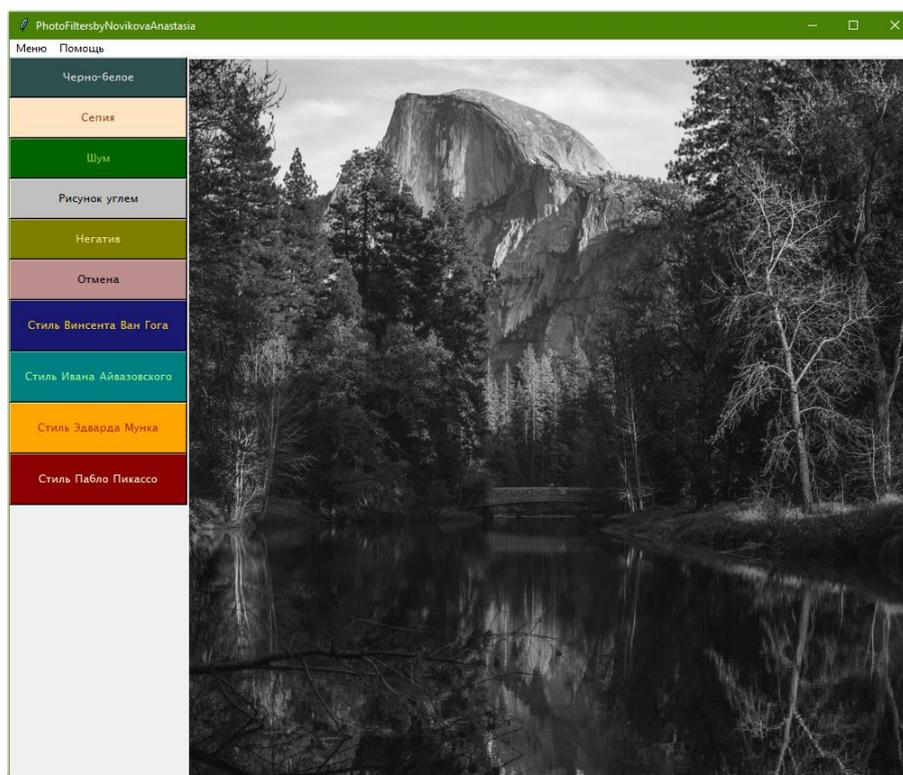


Рисунок 10 – Результат применения фильтра для получения черно-белого изображения

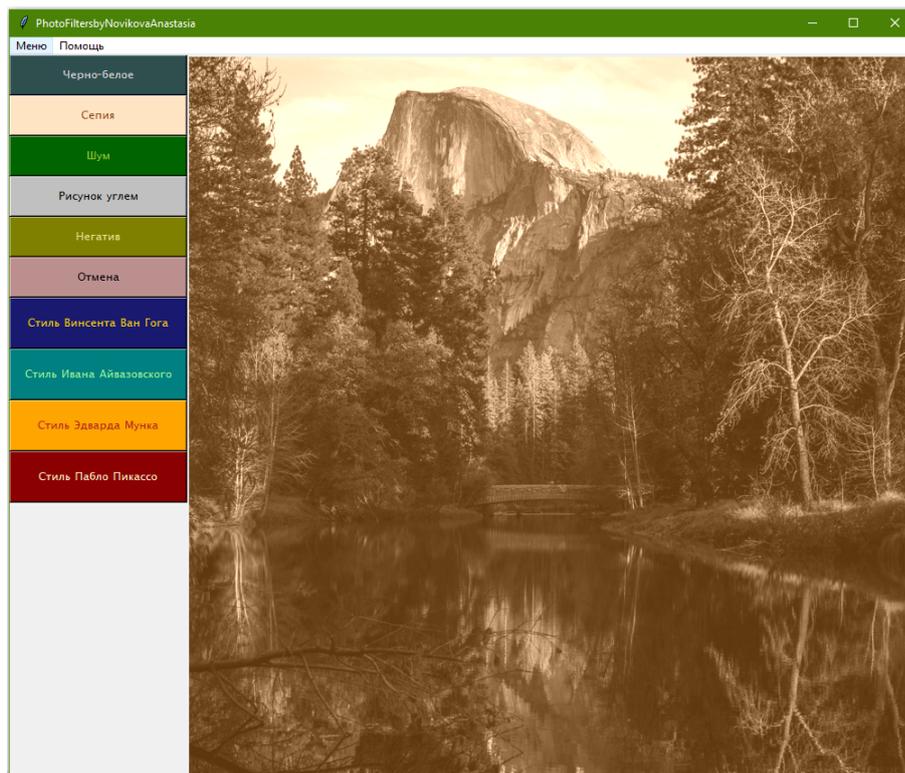


Рисунок 11 – Результат применения фильтра для получения изображения с эффектом сепии

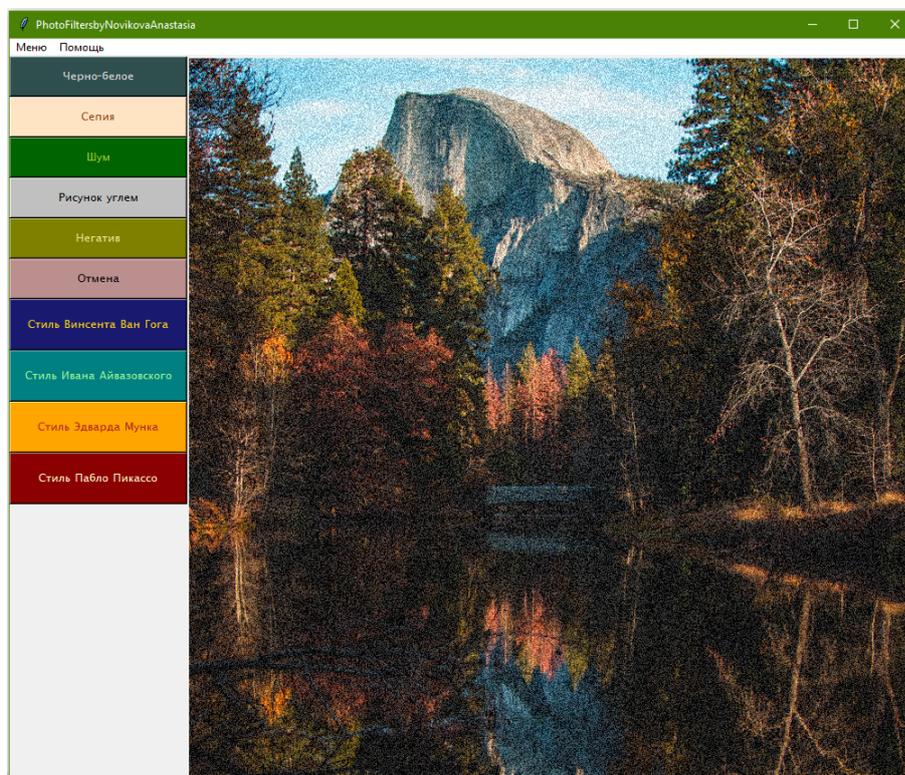


Рисунок 12 – Результат применения фильтра для получения изображения с эффектом зернистости

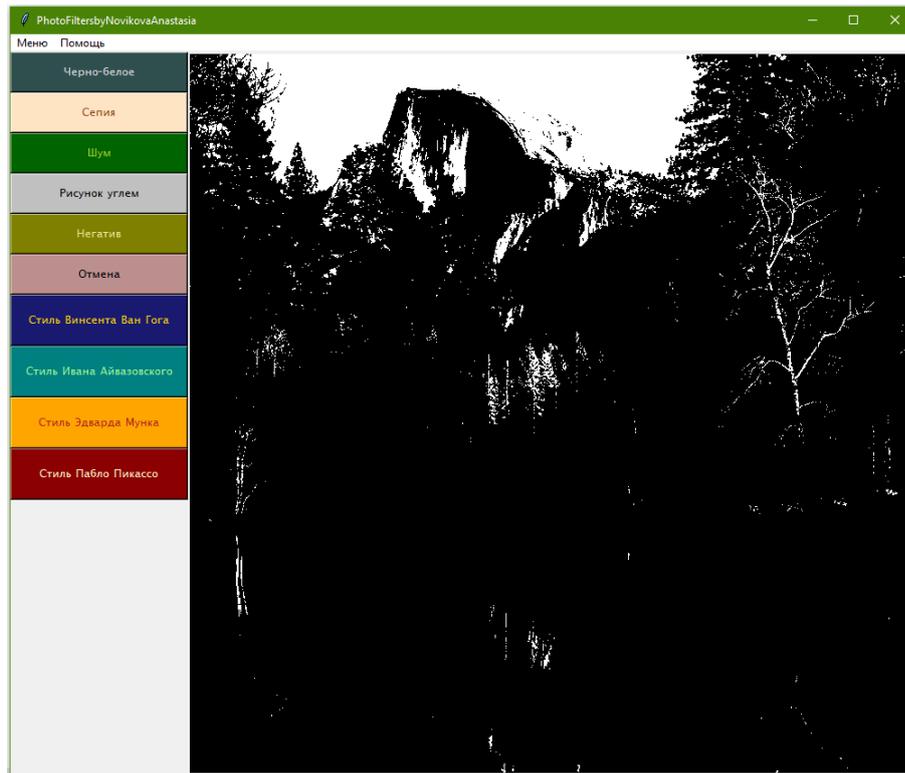


Рисунок 13 – Результат применения фильтра для получения изображения, которое выглядит будто нарисованная углем картина

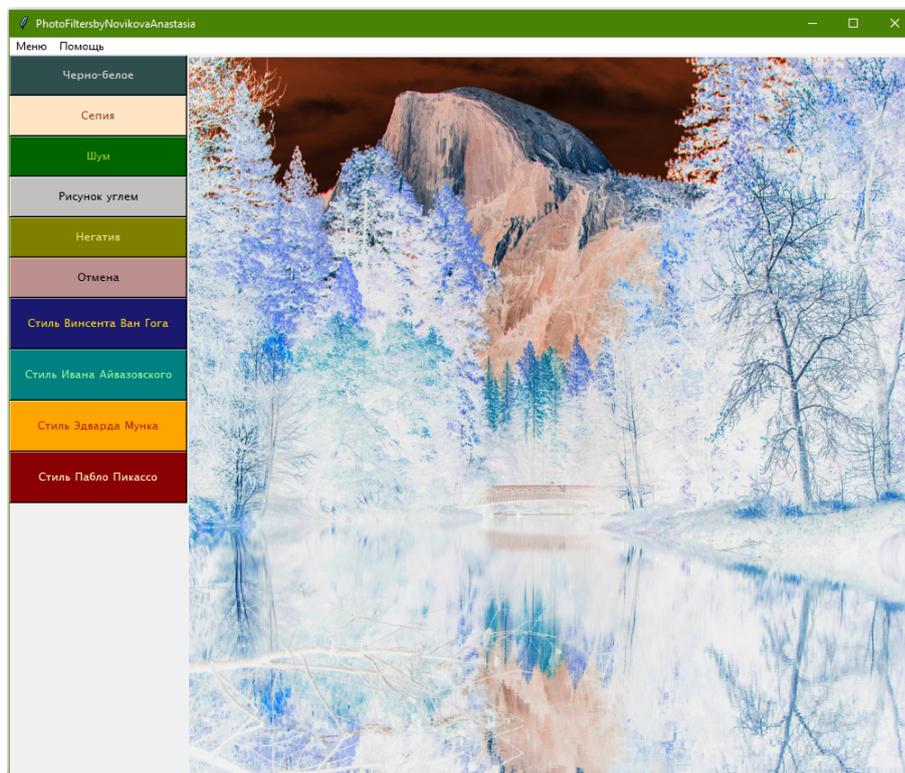


Рисунок 14 – Результат применения фильтра для получения изображения с эффектом негатива

ЗАКЛЮЧЕНИЕ

В процессе выполнения настоящей работы было реализовано графическое приложение на языке программирования Python. Разработанное приложение позволяет редактировать фотографии. Для любителей классической обработки был разработан набор стандартных фильтров: черно-белый фильтр, сепия, шум, рисунок углем и негатив. Для желающих получить необычные фотографии были созданы фильтры, разработанные на основе нейронного переноса стиля.

Для переноса стиля были взяты работы таких известных художников, как Винсент Ван Гог, Иван Айвазовский, Эдвард Мунк и Пабло Пикассо. Таким образом, выбранная пользователем фотография после наложения соответствующего фильтра станет выглядеть словно картина, написанная одним из великих художников, чей стиль работ сложно не узнать.

Все сформулированные задачи были выполнены. Цель, поставленная в работе, была достигнута.