МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра геоинформатики и информационной безопасности

**ОТЧЕТ ПО ПРАКТИКЕ**

|  |  |
| --- | --- |
| Вид практики: | производственная практика |
|  | (учебная, производственная) |
| Тип практики: | научно-исследовательская работа |

Сроки прохождения практики: с 08.02.2022 г. по 10.06.2022 г.

по направлению подготовки 10.05.03 Информационная безопасность автоматизированных систем

(уровень академического специалитета)

направленность (профиль) «Обеспечение информационной безопасности распределенных информационных систем»

|  |  |
| --- | --- |
| Студент группы | № 6412-100503DРодин В.А. |
| Руководитель практики  от университета | Максимов А.И. |

Дата сдачи 10.06.2022 г.

Дата защиты 10.06.2022 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра геоинформатики и информационной безопасности

**Индивидуальное задание на практику**

Студенту группы № 6412-100503DВ.А.Родину

|  |
| --- |
| Направление на практику оформлено приказом по университету от 20.01.2020 № 28-ПР на кафедру геоинформатики и информационной безопасности Самарского университета |
| (наименование профильной организации или структурного подразделения университета) |

|  |  |  |
| --- | --- | --- |
| **Планируемые результаты освоения образовательной программы (компетенции)** | **Планируемые результаты практики** | **Содержание задания** |
| ОК-1 | Знать:  основные направления, проблемы, теории и методы философии, содержание современных философских дискуссий по проблемам общественного развития  Уметь:  использовать положения и категории философии для оценивания и системного анализа различных социальных тенденций, фактов и явлений и моделирования процессов в научной  деятельности; Владеть: - навыками анализа текстов, имеющих философское содержание | Научиться использовать методы философского анализа для изучения различного вида литературы |
| ОК-2 | Знать:  базовые экономические понятия, законы функционирования экономики и поведения экономических агентов, показатели макроэкономического уровня развития страны, экономические показатели, используемые для оценки производственно- хозяйственной деятельности промышленных предприятий Уметь:  использовать понятийный аппарат экономической науки для описания экономических и финансовых процессов  Владеть: навыками использования экономических знаний в сфере личных финансов и профессиональной деятельности. | Пользоваться знаниями экономических понятий и законов в целях при изучении экономических аспектов поставленных задач |
| ОК-3 | Знать:  закономерности и этапы исторического процесса, основные исторические факты, даты, события и имена исторических деятелей России; основные события и процессы отечественной истории в контексте мировой истории  Уметь:  критически воспринимать, анализировать и оценивать историческую информацию, факторы и механизмы исторических изменений  Владеть: навыками анализа причинно-следственных связей в развитии российского государства и общества; места человека в историческом процессе и политической организации общества; навыками уважительного и бережного отношения к историческому наследию и культурным традициям. | Уметь применять знания истории и навыки анализа причинно-следственных связей для решения поставленных задач |
| ОК-5 | Знать:  основные закономерности взаимодействия человека и общества, специфику профессиональной деятельности; основы социологии, структуру общества и социальных институтов; основные этические понятия, историю этических учений, современное положение в сфере этического знания; основные понятия культурологии, типологию культур. Уметь:  определять место и роль профессии в социальной сфере, взаимосвязь с другими профессиями; создавать и поддерживать высокую мотивацию к выполнению профессиональной деятельности; ориентироваться в этической проблематике; выявлять основные черты и особенности культурно-исторических ценностей.  Владеть: методами выявления мотивов социального поведения; технологиями анализа и прогноза социокультурных процессов для решения практических профессиональных проблем. | Пользоваться знаниями о закономерностях взаимодействия человека и общества для решения профессиональных проблем при анализе разрабатываемого метода |
| ОК-9 | Знать:  основные средства и методы физического воспитания Уметь:  выбирать и применять методы и средства физической культуры для совершенствования основных физических качеств Владеть: навыками использования методов и средств физической культуры для обеспечения полноценной социальной и профессиональной деятельности. | Выполнять физические упражнения в свободное от работы время для предотвращения проблем со здоровьем |
| ОПК-6 | Знать: правила поиска и содержание основных нормативно-правовых документов регулирующих работу в области обеспечения информационной безопасности информационных систем Уметь:  разрабатывать локальные и объектовые нормативно-правовые документы для обеспечения нормативно-правового сопровождения работ по обеспечению информационной безопасности на предприятии. Владеть: навыками систематизации и выбора необходимой нормативно-правовой информации согласно поставленным задачам в области обеспечения информационной безопасности автоматизированных систем. | Использовать знания поиска основных нормативно-правовых документов, регулирующих работу в области обеспечения информационной безопасности для выбора необходимой нормативно-правовой информации при решении поставленных задач |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата выдачи задания: | 08.02.2022 г. | | | | |
| Срок предоставления на кафедру отчета о практике: | | | | 10.06.2022 г. | |
| Руководитель практики от | | |  | |  |
| университета, кафедры ГИиИБ | | |  | | Максимов А.И. |
|  | | | (подпись) | |  |
|  | | |  | |  |
| Задание принял к исполнению | | |  | |  |
| студент группы № 6412-100503D | |  | | | В.А. Родин |
|  | | (подпись) | | |  |

**Рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| Дата (период) | Содержание задания | Результаты практики |
| 08.02.2021 – 22.02.2021 | Изучить существующие методы детектирования объектов на изображении | Были изучены основные методы детектирования объектов на изображении, описан принцип работы каждого из методов |
| 22.02.2021 – 08.03.2021 | Изучить существующие методы детектирования и локализации лица на изображениях | Были изучены основные методы детектирования и локализации лица на изображениях, описан принцип работы каждого из методов |
| 08.03.2022 – 08.04.2022 | Изучить эффективность применения переноса стиля для задач сопоставления лиц на натуралистическом портретном изображении и наброске. | Эффективность переноса стиля была изучена, по результатам экспериментального исследования выяснилось, что перенос стиля лишь незначительно сокращает меру близости энкодингов лиц – чуть менее, чем в 1,1 раз |
| 27.04.2021 –  10.06.2021 | Разработать автоматизированный метод сопоставления натуралистического портретного изображения и композитного портрета | Описано текущее программное обеспечение и перспективы его дальнейшего развития. |

|  |  |  |
| --- | --- | --- |
| Руководитель практики от |  |  |
| университета, ассистент кафедры ГИиИБ |  | Максимов  А.И. |
|  | (подпись) |  |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 8](#_Toc104901329)

[ВЫПОЛНЕНИЕ ЗАДАНИЯ 8](#_Toc104901330)

[1 Введение 8](#_Toc104901331)

[1.1 Актуальность проблемы 8](#_Toc104901332)

[1.2 Цель и задачи исследования 8](#_Toc104901333)

[1.3 Основные понятия 9](#_Toc104901334)

[2 Обзор существующих методов по тематике исследования 9](#_Toc104901335)

[2.1 Задача детектирования объектов на изображении 10](#_Toc104901336)

[3 Примерный процесс разработки 11](#_Toc104901337)

[4 Задача детектирования и локализации лица 11](#_Toc104901338)

[4.1.1 Snakes 12](#_Toc104901339)

[4.1.2 Модель сопоставления шаблонов 12](#_Toc104901340)

[4.1.3 Deformable Parts Model 13](#_Toc104901341)

[4.1.4 модель распределения точек (PDM) 14](#_Toc104901342)

[5 Задача распознавания человека сравнением натуралистического портретного изображения с композитным портретом 15](#_Toc104901343)

[5.1 Метод распознавания натуралистического изображения лица и наброска использующий контрольную сумму локального градиента (LGCS) 15](#_Toc104901344)

[5.2 Метод основанный на самоподобие лица 16](#_Toc104901345)

[5.3 Метод основанный на признаках 17](#_Toc104901346)

[5.4 Метод основанный на признаках 18](#_Toc104901347)

[6 Перенос стиля изображения при помощи нейронной сети 19](#_Toc104901348)

[7 Разработка программы и исследование методов распознавания лиц 21](#_Toc104901349)

[ЗАКЛЮЧЕНИЕ 23](#_Toc104901350)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc104901351)

[ПРИЛОЖЕНИЕ А 25](#_Toc104901352)

[**ОТЗЫВ О ПРОХОЖДЕНИИ ПРАКТИКИ** 36](#_Toc104901353)

О Т Ч Е Т

о выполнении индивидуального задания

по научно-исследовательской работе

# ВВЕДЕНИЕ

При прохождении практики по научно-исследовательской работе, руководителем были поставлены следующие задачи:

1. изучить существующие методы детектирования объектов на изображении;
2. изучить существующие методы детектирования и локализации лица на изображениях;
3. разработать автоматизированный метод сопоставления натуралистического портретного изображения и композитного портрета.

Задания необходимо было выполнять последовательно в течение всего времени практики, предоставляя руководителю промежуточные отчеты.

# ВЫПОЛНЕНИЕ ЗАДАНИЯ

## 1 Введение

### 1.1 Актуальность проблемы

Для выполнения поставленных руководителем задач был проведен анализ тем и обоснована актуальность выбранной темы в области защиты информации в современном обществе. Определены цель и задачи исследования.

На данный момент существует большое количество методов идентификации человека по лицу, эти методы успешно внедряются повсеместно – от аэропортов до домофонов жилых кварталов. Однако у этой задачи есть частный случай – когда фотографии человека нет, но есть его фоторобот. Статей на тему сравнения натуралистического изображения и фоторобота не так много, а систем, реализующих это и вовсе мало. Поэтому эта задача довольно актуальна и имеет научную новизну.

### 1.2 Цель и задачи исследования

Целью исследования стала разработка алгоритма распознавания человека сравнением натуралистического портретного изображения с композитным портретом. Задачи исследования:

1. изучение уже существующих алгоритмов и их анализ с точки зрения успешности результата распознавания (правильным ли оказался результат распознавания);
2. создание нового алгоритма, опережающего уже существующие в данном компоненте.

Объектом исследования стали наброски и фотографии человека. Предмет исследования – наброски и фотографии одних и тех же людей взятые из базы данных.

### 1.3 Основные понятия

Изображение – это ограниченная функция двух пространственных переменных, заданная на ограниченной прямоугольной плоскости[1].

Изображение описывается выражением

, (1.1)

Функциюв дальнейшем будем называть изображением, где - функция распределения яркости, - функция, описывающая спектральную чувствительность, - длина волны светового освещающего изображения излучение, а - пространственные координаты изображения.

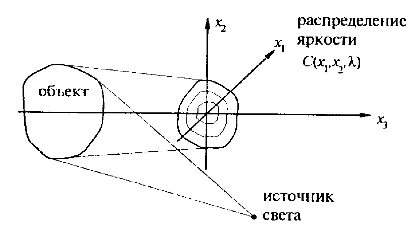


Рисунок 1 – формирование изображения объекта, освещенного источником света

Видеоданные могут быть разного типа – видеопоток и видеопоследовательность. Видеопоток – упорядоченная последовательность изображений, полученных с одной камеры через небольшие промежутки времени. Видеопоток подразумевает обработку в реальном времени. Видеопоследовательность конечна, ее можно обрабатывать целиком[1].

Задача классификации изображений состоит в том, чтобы принять решение о принадлежности образа к тому или иному классу на основе анализа вычисленных признаков [1].

1. Обзор существующих методов по тематике исследования

Исследование будет проводиться в нескольких областях:

1. методы обработки изображений;
2. теория распознавания образов;
3. машинное обучение.

### Задача детектирования объектов на изображении

Задача детектирования объектов на изображении - задача машинного обучения, в рамках которой выполняется определение наличия или отсутствия объекта определённого класса на изображении, нахождение границ этого объекта в системе координат пикселей исходного изображения[2].

Выделяют такие задачи детектирования объектов на изображении как задача семантической сегментации - задача, в которой на вход модели подаётся изображение, а на выходе для каждому пикселю сопоставляется метка принадлежности этого пикселя к определённой категории, задача классификации с локализацией - задача, в которой в дополнение к предсказанию метки категории класса определяется рамка, ограничивающая местоположение экземпляра одиночного объекта на картинке и задача детекции объектов - задача, в рамках которой необходимо выделить несколько объектов на изображении посредством нахождения координат их ограничивающих рамок и классификации этих ограничивающих рамок из множества заранее известных классов[2].

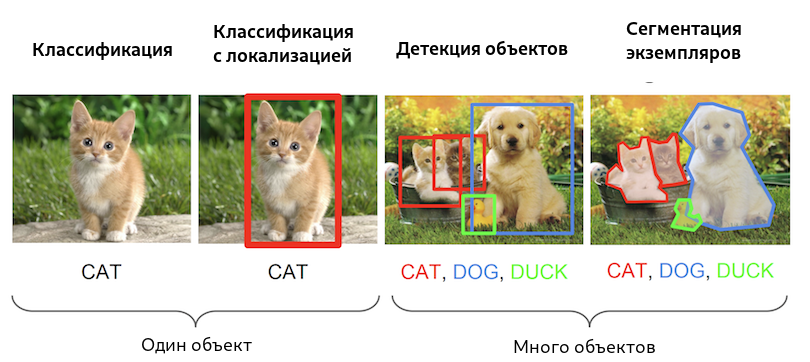


Рисунок 1 – Различия между задачами нахождения объектов на изображении

## 3 Примерный процесс разработки

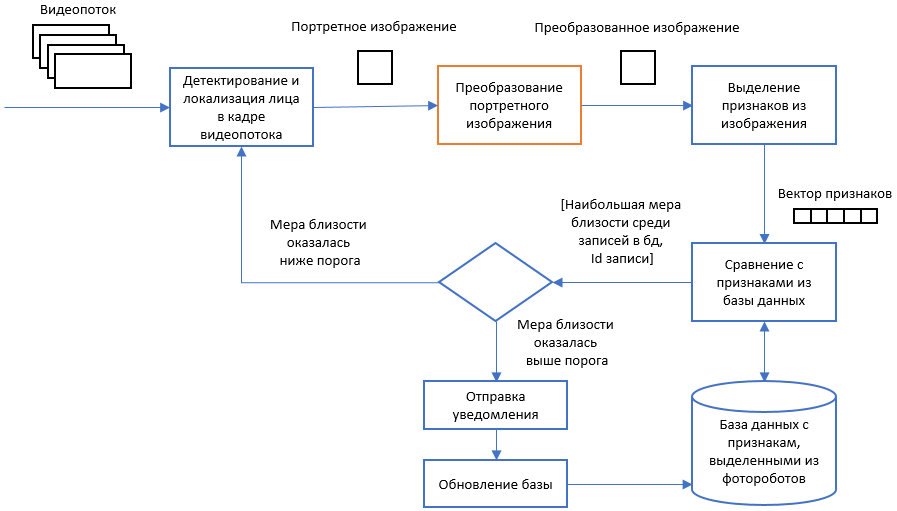


Рисунок 2 – примерный процесс разработки программы

Процесс разработки программы можно поделить на несколько этапов:

1. детектирование и локализация лица в кадре видеопотока;
2. преобразование портретного изображения;
3. выделение признаков из изображения;
4. сравнение с признаками из базы данных;
5. отправка уведомления, если мера близости оказалась выше порога или повторение шагов 1 – 4 с использованием других алгоритмов на шагах 1 – 3 иначе.

## 4 Задача детектирования и локализации лица

Решением задачи детектирования и локализации лица является область/области изображения, на который изображено лицо/несколько лиц. Основными проблемы, возникающие при решении этой задачи это окклюзия, освещение и сложный фон. Именно поэтому существуют различные алгоритмы детектирования и локализации лица, по-разному решающие эти проблемы.

Существующие алгоритмы детектирования и локализации лица можно разделить на две части:

1. алгоритмы, основанные на признаках;
2. алгоритмы, основанные на изображениях.

4.1 Методы, основанные на признаках

Методы, основанные на признаках подразделяются на три подраздела:

1) режим активной формы;

2) анализ низкого уровня;

3) анализ признаков.

Системы, использующие режим активной формы генерируют маску на основе найденных черт лица таких как нос, рот и уши. Сгенерированные маски можно изменять. Лучший результат можно получить, тренируя систему большим количеством изображений. Данный метод можно разделить на четыре группы – snakes[2], модель сопоставления шаблонов (DTM)[3], Deformable Parts Model (DPM)[4] и модель распределения точек (PDM)[5].

### 4.1.1 Snakes

Этот метод обычно используется для определения границ головы, он относительно нечувствителен к шуму, так как интегральный оператор, использующийся в них для как внутренних, так и внешних энергетических функций является фильтром шума.

Энергетическая функция E(C) хорошо выполняет задачу сегментации и описывается выражением

, где и - функции внутренней и внешней энергии соответственно.

Как правило, этот метод способен определять границы объектов, но обладает некоторыми ограничениями, например, контуры часто попадают в ловушку ложных объектов изображения, а также плохо подходят для нахождения невыпуклых объектов.

### 4.1.2 Модель сопоставления шаблонов

В отличие от метода snakes, задача метода модели сопоставления шаблонов состоит не только в обнаружении границ, но и в обнаружении таких черт лица как глаза, рот, брови, нос и уши. Эта модель работает путем формирования деформируемых форм лица, что достигается заранее заданными формами, которые могут быть как многоугольными шаблонами, так и иерархическими шаблонами.



Рисунок 3 - многоугольный шаблон человеческого лица

На рисунке 3 видно, что лицо формируется из нескольких треугольников, каждый из которых деформирован, чтобы скорректировать общую форму лица.

Модель обеспечивает лучшее выделение нужных черт за счет объединения локальной информации с глобальной, но, как и модель snakes, требует чрезмерное время обработки.

### 4.1.3 Deformable Parts Model

DPM использует графическую структуру, предложенную Fischler and Elschlager et al. [6] для распознавания объектов. Графическую структуру можно разделить на две части: частичные фильтры и корневой фильтр. DPM обычно используется для распознавания лиц, а также для обнаружения лиц в комиксах. В этом методе маска лица формируется путем индивидуального моделирования отдельных частей (глаз, носа и т.д.). Между этими частями устанавливается набор геометрических ограничений, обычно описывающих расстояние между глазами, носом и т.д. Эти ограничения можно представить в виде пружин.

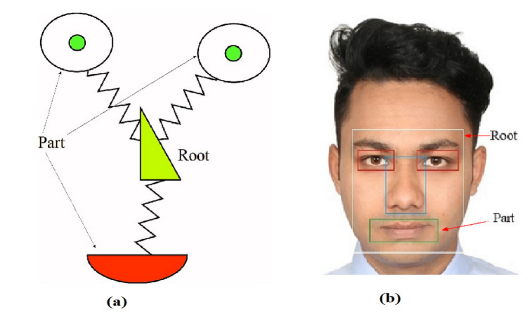


Рисунок 4 – (а) – маска лица, в которой такие части как глаза, нос и рот соединены геометрическими ограничениями - пружинами. (b) – графическая структура, спроецированная на реальное человеческое фото, с четким указанием частичных и корневых фильтров.

### 4.1.4 модель распределения точек (PDM)

В PDM форма лица описывается точками. Метод основан на ориентирах, где ориентир – это аннотации изображения к любой заданной фигуре из изображениях обучающего набора. Форма лица в PDM формируется путем установки ориентиров на форму лица в наборе обучающих изображений. Модель обычно строится с формой лица, имеющей формы глаз, ушей, носа и других элементов лица.

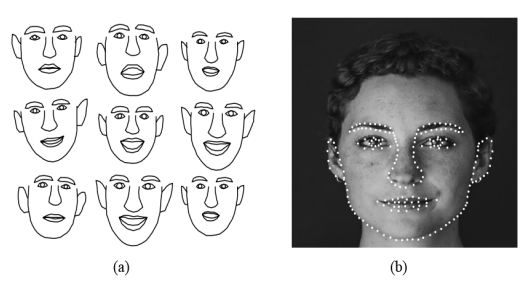


Рисунок 5 – (a) – типичные обучающие формы лица. (b) - модельные точки, проецируемые на обучающее изображение с лицом, которые создают основные формы лица.

Метод может использоваться и для поиска объемных(3D) данных. Одним из недостатков является сложность построения обучающего набора путем указания границ и черт лица ввиду неизбежно тяжелой работы и возникновения ошибок.

## 5 Задача распознавания человека сравнением натуралистического портретного изображения с композитным портретом

Задача распознавания человека сравнением натуралистического портретного изображения с композитным портретом состоит в том, чтобы определить, один и тот же человек на фотографии и на эскизе или нет.

Существуют следующие методы решения данной задачи:

1. метод использующий контрольную сумму локального градиента

LGCS [7];

1. метод основанный на самоподобии лица[8];
2. метод на основе признаков[9];
3. метод, использующий гибрид двух методов[10];

### 5.1 Метод распознавания натуралистического изображения лица и наброска использующий контрольную сумму локального градиента (LGCS)[7]

Метод состоит из следующих этапов:

1) набросок и фотоизображения выравниваются в соответствии с координатами глаз, обрезаются и изменяются в размерах;

2) система преобразовывает эскиз в градиентное изображение, а затем в LGCS изображение

3) LGCS изображение наброска сопоставляется с LGCS изображениями из базы данных и на выход дается результат из четырех лучших совпадений в соответствии с расстоянием Евклида. Это так называемая система распознавания эскизов лиц и фотографий (FSPR).

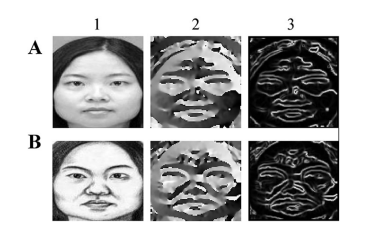


Рисунок 6 – пример преобразования фотографии - строка А и эскиза – строка B в градиентное изображение – столбец 2 и в LGCS изображение – столбец 3

### 5.2 Метод основанный на самоподобие лица[8]

В данном методе изображения проходят предварительную обработку, в результате которой как эскизы, так и фотографии поворачиваются по отношении к определенным вручную координатам глаз, и масштабируются таким образом, чтобы расстояние между глазами составляло 25 пикселей. Область размером 100×100 обрезается таким образом, чтобы глаза были зафиксированы в 48 строке изображения.

Далее изображение подвергается обрабатывается с помощью фильтра Difference-of-Gaussian (DoG). Результирующее изображение D(x,y) получается с помощью свертки исходного изображения I(x,y) с фильтром DoG.

D(x, y)=(G(x, y, σ2) − G(x, y, σ1)) ∗ I(x, y) , (2)

где σ – стандартное отклонение в фильтрах Гаусса.



Рисунок 7 – нормализация и предварительная обработка фотографий и эскизов. Оригинальные изображения (верхняя часть), нормализованные изображения (центр), обработанные изображения (низ)

Далее для фотографий и эскиза вычисляется дескриптор Facial Self Similarity (FSS). Вычисление состоит из двух этапов:

1. вычисление поверхности самоподобия;
2. преобразование поверхности самоподобия в полярную гистограмму.

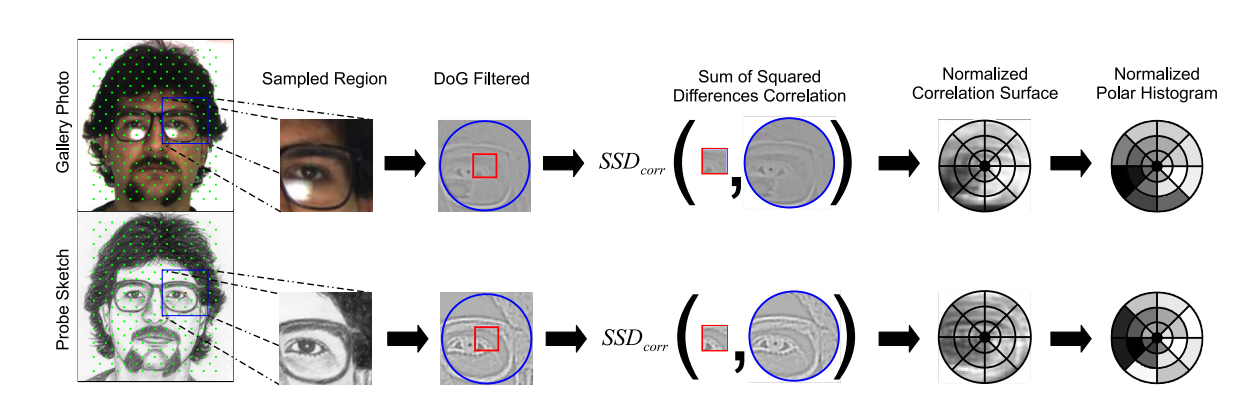


Рисунок 8 – пример расчета FSS дескриптора для фотографии и эскиза

Далее изображения сопоставляются с помощью евклидова расстояния.

Данный метод превосходит такие методы как PCA Eigenface, FaceVACS, а также метод дискриминантного анализа по локальным признакам таким как SIFT и SIFT+MLBP, что показывает, что FSS лучше справляется с неоднородной природой эскизов и фотографий. Особенностью метода является то, что координаты глаз определяются вручную.

### 5.3 Метод основанный на признаках[9]

Метод включает в себя следующие этапы:

1. предварительная обработка фотографий и эскизов;
2. извлечение характеристик обоих изображений с помощью гибрида дескрипторов Histogram of Oriented gradient (HOG)[11] и Gray Level Co-Occurrence Matrix (GLCM)[12].



Рисунок 9 - (a), (b), (c) – оригинальный эскиз, его полутоновое изображение и изображение обработанное медианным фильтром, (d), (e), (f) – исходная фотография, ее полутоновое изображение и обработанное медианным фильтром полутоновое изображение

Эскизы и фотографии сопоставляются с помощью евклидова расстояния.

В предварительной обработке изображения преобразовываются из цветного в черно-белый, затем удаляются шумы с помощью медианного фильтра.

HOG вычисляется для каждого изображения, формируется вектор признаков, который используется для сопоставления.

GLCM используется для извлечения информации о текстуре изображения. Преимущество вычисления GLCM заключается в том, что он дает как вариацию значений пикселей, так и их относительное положение.

Преимущество данного метода – использование двух дескрипторов, вместо одного. Предлагаемый метод дает лучшие результаты по сравнению с другими современными методами и может быть улучшен в будущем.

### 5.4 Метод основанный на признаках[10]

Метод демонстрирует возможность сравнивать фотографии с эскизами несколькими способами:

1. с помощью дескрипторов SIFT – метод прямого сопоставления;
2. сравнивая расстояния с обучающим набором пар фотография/эскиз – метод общего сопоставления;
3. используя гибрид методов 1 и 2.

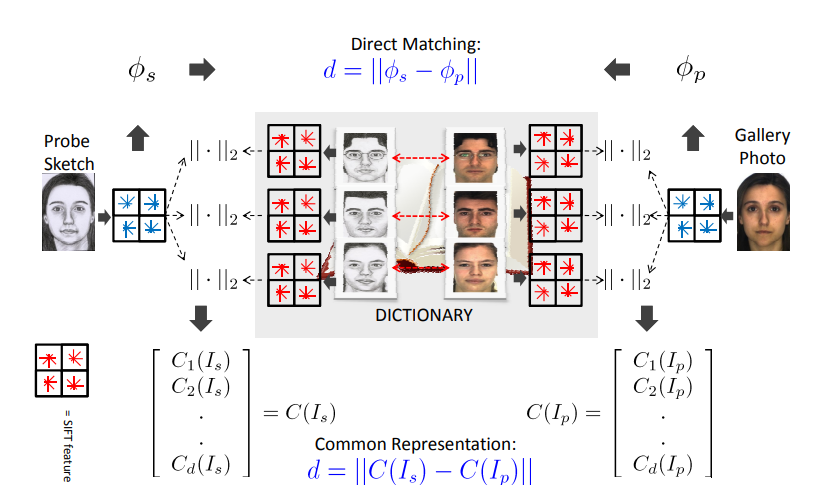


Рисунок 10 – процесс сравнения эскиза с фотографией методами прямого сопоставления (верх) и метода общего представления (низ)

Первый шаг – вычисление SIFT – представления для каждого изображения. Прямое сопоставление выполняется путем вычисления расстояния SIFT – представления между эскизом и фотографией. Для общего представления эскиз описывается как

d-мерный вектор, где d – произведение числа предметов в словаре и числа участков, отобранных для создания SIFT – характеристик. Компоненты вектора d – это расстояния L2 от p выборочных SIFT – дескрипторов эскиза до тех же дескрипторов для каждого из n эскизов в обучающем наборе. Затем этот же процесс применяется к фотографиям из галереи, при этом сравнение идет с фотографиями из обучающего набора.

Метод может быть улучшен, если вместо дескриптора SIFT использовать другие дескрипторы, например, SURF, ORB или BRIEF.

## 6 Перенос стиля изображения при помощи нейронной сети

Алгоритм переноса стиля [13] преобразует полученное на вход изображение в соответствии с выбранным стилем (цветовым наполнением, текстурой), полученным из второго изображения. В данной работе для переноса стиля используется архитектура нейронной сети, предложенная в работе [14]. Данная сеть состоит из трех подсетей – подсети, непосредственно осуществляющей перенос стиля, подсети, предсказывающей эмбеддинги для подсети переноса стиля и подсети потерь. Первая подсеть осуществляет преобразование входного изображения на основе полученных от второй подсети данных. Архитектура данной сети представлена на рисунке 11.

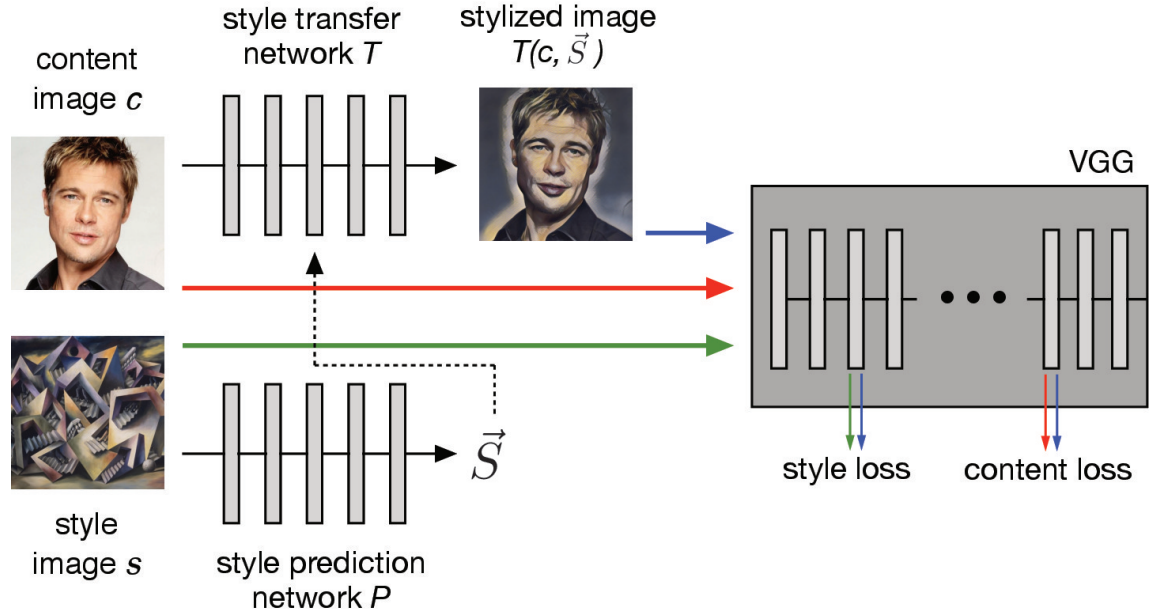


Рисунок 11 - архитектура сети [14], использованной для переноса стиля в данном исследовании

Для исследования был самостоятельно создан набор тестовых данных, представляющий из себя набор фронтальных портретных фото на белом фоне и набросков тех же людей. Для натуралистических изображений был произведен перенос стиля набросков при помощи [14]. На рисунке 12 продемонстрированы примеры таких изображений.

а) б) в)

Рисунок 12 - примеры изображений тестового набора данных – а) портретное натуралистическое изображение, б) набросок того же человека, в) портретное изображение после переноса стиля наброска. Мера близости между изображениями а) и б) – 0,5821, между в) и б) – 0,5795.

Исследование производилось следующим образом – из пары изображений одного и того же человека вычислялись энкодинги лиц при помощи метода, описанного в работе [15], после чего вычислялась мера близости (евклидово расстояние) между получившимися энкодингами. Расстояние усреднялось по всему тестовому набору. Таким образом были получены средние значения расстояния между энкодингами для пар вида (портретное фото, набросок) и (портретное фото с перенесенным стилем наброска, набросок). Результаты исследования приведены в таблице 1:

|  |  |  |
| --- | --- | --- |
| **№** | **Пары изображений** | **Среднее расстояние** |
| 1 | Портретные изображения и наброски без переноса стиля | 0,612 |
| 2 | Портретные изображения с переносом стиля набросков и наброски | 0,5709 |

Как видно из приведенных в таблице 1 значений, использование переноса стиля позволило сократить меру близости энкодингов изображений в среднем в 1,07 раз. Из чего можно сделать вывод, что использование методов переноса стиля лишь незначительно помогает при решении задачи распознавании лиц по наброску с использованием [15].

В дальнейшем планируется развивать данное исследование и рассмотреть как другие методы распознавания лиц, так и другие методы предобработки изображений.

## 7 Разработка программы и исследование методов распознавания лиц

Программное обеспечение написано на языке Python. Для разработки программы была использована ранее рассмотренная в пункте 6 библиотека tensorflow и библиотека arcface которая для распознавания лиц использует алгоритм, описанный в [16]

На данный момент реализовано несколько модулей

1. utils – этот модуль содержит в себе необходимые для подсчета эмбеддингов функции с помощью arcface так и tensorflow(использовался для проведения эксперимента с переносом стиля)
2. tests – модуль в котором были произведены тесты переноса стиля, а также распознавания лица с помощью arcface и tensorflow;
3. pipeline\_implementation – модуль, в котором из видеопотока с помощью arcface происходит детектирование лица. Далее для обнаруженного лица подсчитываются эмбедденги и находится наиболее близкое по расстоянию между эмбедденгами изображение из тестового набора данных;
4. images – модуль с тестовыми изображениями и набросками;
5. output\_embeds – модуль в котором хранятся подсчитанные ранее эмбедденги для тестового набора изображений.

Работа программы заключается в следующем: из видеопотока с помощью arcface происходит детектирование лица, далее для обнаруженного лица подсчитываются эмбедденги и находится наиболее близкое по расстоянию между эмбедденгами изображение из тестового набора данных.

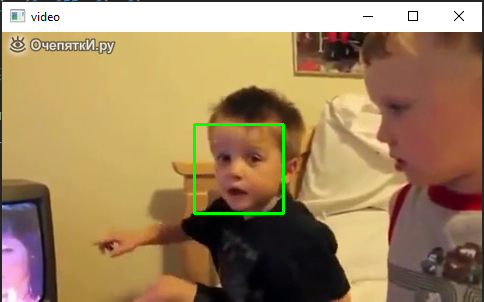


Рисунок 13 – пример детектирования лица из видеопотока

В дальнейшем планируется заменить работу с локальными файлами на работу с базой данных, рассмотреть различную предобработку изображений и рассмотреть другие методы распознавания лиц.

# ЗАКЛЮЧЕНИЕ

В течение практики по научно-исследовательской работе успешно выполнены поставленные задачи: были проанализированы темы в области обработки изображений. Определена и обоснована актуальность выбранной темы. Определена цель и задачи исследований. Определен объект и предмет исследования. Были изучены некоторые методы детектирования лица на изображении. Также была рассмотрена задача распознавания человека сравнением натуралистического портретного изображения с композитным портретом и были исследованы основные решения данной задачи. Была исследована часть задачи распознавания лиц по наброску, а именно - эффективность применения переноса стиля для задач сопоставления лиц на натуралистическом портретном изображении и наброске. По результатам экспериментального исследования выяснилось, что перенос стиля лишь незначительно сокращает меру близости энкодингов лиц – чуть менее, чем в 1,1 раз.

За время прохождения практики освоены необходимые компетенции, в частности овладел практическими навыками детектирования лица на изображении.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1.Книга Методы компьютерной обработки изображений (Коллектив авторов под редакцией В. А....) - большая электронная библиотека [Electronic resource]. URL: https://bookree.org/reader?file=578401 (accessed: 28.02.2022).

2. Kass M., Witkin A., Terzopoulos D. Snakes: Active contour models // Int J Comput Vision. 1988. Vol. 1, № 4. P. 321–331.

3. Yuille A.L., Hallinan P.W., Cohen D.S. Feature extraction from faces using deformable templates // Int J Comput Vision. 1992. Vol. 8, № 2. P. 99–111.

4. Cootes T., Taylor C. Active Shape Models - “smart snakes” // BMVC. 1992.

5. Lanitis A., Taylor C.J., Cootes T.F. Automatic tracking, coding and reconstruction of human faces, using flexible appearance models // Electronics Letters. IET Digital Library, 1994. Vol. 30, № 19. P. 1587–1588.

6. Fischler M.A., Elschlager R.A. The Representation and Matching of Pictorial Structures // IEEE Transactions on Computers. 1973. Vol. C–22, № 1. P. 67–92.

7. Roy H., Bhattacharjee D. Face sketch-photo recognition using local gradient checksum: LGCS // International Journal of Machine Learning and Cybernetics. 2017. Vol. 8.

8. Khan Z., Hu Y., Mian A. Facial Self Similarity for Sketch to Photo Matching. 2012. P. 1–7.

9. Dalal S., Vishwakarma V., Kumar S. Feature-based Sketch-Photo Matching for Face Recognition // Procedia Computer Science. 2020. Vol. 167. P. 562–570.

10. Klare B., Jain A. Sketch to Photo Matching: A Feature-based Approach // Proceedings of SPIE - The International Society for Optical Engineering. 2010.

11. Shu C., Ding X., Fang C. Histogram of the Oriented Gradient for Face Recognition \* // Tsinghua Science & Technology. 2011. Vol. 16. P. 216–224.

12. Shi Z., Chen H., Xuanjing S. Splicing Image Forgery Detection Using Textural Features Based on the Gray Level Co-occurrence Matrices // IET Image Processing. 2016. Vol. 11.

13. Gatys L.A. A Neural Algorithm of Artistic Style / L.A. Gatys, A.S. Ecker, M. Bethge // Journal of Vision. – 2016. – Vol.16. – №326.

14. Ghiasi G. Exploring the structure of a real-time, arbitrary neural artistic stylization network/ G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, J. Shlens // Proceedings of the British Machine Vision Conference (BMVC). – 2017.

15. Amos B. Openface: A general-purpose face recognition library with mobile applications/ B. Amos, B. Ludwiczuk, M. Satyanarayanan // CMU-CS-16-118, CMU School of Computer Science, Tech. Rep. – 2016.

16. Deng J. et al. ArcFace: Additive Angular Margin Loss for Deep Face Recognition: arXiv:1801.07698. arXiv, 2019.

# ПРИЛОЖЕНИЕ А

**Исходный код программы**

Модуль pipeline\_implementation

Файл main.py

import os.path

import cv2

from arcface.lib import ArcFaceModel

import utils.my\_arcface.main

import numpy as np

def get\_best\_distanse(embeds):

# больше точно не будет

best\_distance = 100

path\_best\_embeds = -1

"C:\CompositePortraitRecongnition\output\_embeds\1.npy"

for i in range(1, 19):

name = str(i) + ".npy"

path = os.path.join("C:\CompositePortraitRecongnition", "output\_embeds", name)

embeds1 = np.load(path)

diff = np.subtract(embeds1, embeds)

distance = np.sum(np.square(diff))

if distance < best\_distance:

best\_distance = distance

path\_best\_embeds = path

return best\_distance, path\_best\_embeds

if \_\_name\_\_ == "\_\_main\_\_":

# CONSTANTS

COUNT\_FRAMES = 5

INPUT\_SIZE = 112

# почему-то если путь к конфигу не абсолютный, то его не воспринимают и это все не работает

cascPath = "C:\\CompositePortraitRecongnition\\configs\\cascade.xml"

faceCascade = cv2.CascadeClassifier(cascPath)

# for calculating embeddings

input\_size = INPUT\_SIZE

model = ArcFaceModel(size=input\_size,

backbone\_type='ResNet50',

training=False)

# video\_capture = cv2.VideoCapture(0)

video\_capture = cv2.VideoCapture('C:\\CompositePortraitRecongnition\\video\\test.mp4')

count = 0

while True:

# Capture frame-by-frame

ret, frame = video\_capture.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray, 1.3, 5)

# Draw a rectangle around the faces

for (x, y, w, h) in faces:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imwrite(f"C:\\CompositePortraitRecongnition\\dataset\\frame{count}.jpg", gray[y:y + h, x:x + w])

count += 1

embeds = utils.my\_arcface.main.calculate\_embeddings(frame, model, input\_size)

best\_distance, path\_best\_embeds = get\_best\_distanse(embeds)

print(best\_distance, path\_best\_embeds)

# Display the resulting frame

cv2.imshow('Video', frame)

if count >= COUNT\_FRAMES:

break

# to stop

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# When everything is done, release the capture

video\_capture.release()

cv2.destroyAllWindows()

Модуль tests

Файл testing\_arcface\_and\_tensorflow.py

import os

from datetime import datetime

import openpyxl

from arcface.lib import ArcFaceModel

import utils.my\_arcface.main

import utils.tensorflow.style\_transfer

import utils.tensorflow.face\_encoding

if \_\_name\_\_ == "\_\_main\_\_":

# tensorflow model

transfer\_model = utils.tensorflow.style\_transfer.TransferModel(utils.tensorflow.style\_transfer.MODEL\_URL)

headers = ['№ пары изображений',

'Расстояние до переноса стиля tensorflow',

'Расстояние после переноса стиля tensorflow',

'Расстояние до переноса стиля arcface',

'Расстояние после переноса стиля arcface']

# before style\_transfer

arcface\_encodings\_array\_before = []

tensorflow\_encodings\_array\_before = []

# after style\_transfer

arcface\_encodings\_array\_after = []

tensorflow\_encodings\_array\_after = []

# array of indexes of successful operations

successful\_indexes = []

input\_size = 300

# arcface model

model = ArcFaceModel(size=input\_size,

backbone\_type='ResNet50',

training=False)

for i in range(1, 20):

print(f"{datetime.now()}: iteration number {i}")

path0 = os.path.join("../images", "photos", f"photo{str(i + 1)}.png")

path1 = os.path.join("../images", "sketches", f"sketch{str(i + 1)}.png")

try:

tensorflow\_distance = utils.tensorflow.face\_encoding.calculate\_distance(path0, path1)

arcface\_distance = utils.my\_arcface.main.calculate\_distance(path0, path1, input\_size, model)

except IndexError as e:

print(

f"{str(e)} \n не удалось обнаружить лицо на фотографии до переноса стиля") # https://stackoverflow.com/questions/59919993/indexerror-list-index-out-of-range-face-recognition

continue

path\_image\_with\_style = os.path.join("../images", "with\_style", f"{str(i + 1)}.png")

image\_with\_style = transfer\_model.process\_image(path0, path1, path\_image\_with\_style)

try:

new\_tensorflow\_distance = utils.tensorflow.face\_encoding.calculate\_distance(path\_image\_with\_style, path1)

new\_arcface\_distance = utils.my\_arcface.main.calculate\_distance(path\_image\_with\_style, path1, input\_size,

model)

except IndexError as e:

print(f"{str(e)} \n не удалось обнаружить лицо на фотографии после переноса стиля")

continue

tensorflow\_encodings\_array\_before.append(tensorflow\_distance)

arcface\_encodings\_array\_before.append(arcface\_distance)

tensorflow\_encodings\_array\_after.append(new\_tensorflow\_distance)

arcface\_encodings\_array\_after.append(new\_arcface\_distance)

print(f"{datetime.now()}: {tensorflow\_distance}")

print(f"{datetime.now()}: {new\_tensorflow\_distance}")

successful\_indexes.append(i)

book = openpyxl.Workbook()

sheet\_1 = book.create\_sheet("results", 0)

sheet\_1.append(headers)

# в таблицу попадают только успешные преобразования стиля(в которых рассчиталось расстояние до и после)

for i in range(0, len(successful\_indexes)):

sheet\_1.append(

[successful\_indexes[i], tensorflow\_encodings\_array\_before[i], tensorflow\_encodings\_array\_after[i],

arcface\_encodings\_array\_before[i], arcface\_encodings\_array\_after[i]])

book.save("C:\\CompositePortraitRecongnition\\results\\results.xlsx")

# this part calculates average gain for tensorflow

avg\_before\_tensorflow = sum(tensorflow\_encodings\_array\_before) / len(tensorflow\_encodings\_array\_before)

avg\_after\_tensorflow = sum(tensorflow\_encodings\_array\_after) / len(tensorflow\_encodings\_array\_after)

print(

f"{datetime.now()}: avg\_before tensorflow: {avg\_before\_tensorflow}\navg\_after tensorflow: {avg\_after\_tensorflow}")

print(f"{datetime.now()}: improve tensorflow: {avg\_before\_tensorflow / avg\_after\_tensorflow}")

# this part calculates average gain for arcface

avg\_before\_arcface = sum(arcface\_encodings\_array\_before) / len(arcface\_encodings\_array\_before)

avg\_after\_arcface = sum(arcface\_encodings\_array\_after) / len(arcface\_encodings\_array\_after)

print(f"{datetime.now()}: avg\_before arcface: {avg\_before\_arcface}\navg\_after arcface: {avg\_after\_arcface}")

print(f"{datetime.now()}: improve arcface: {avg\_before\_arcface / avg\_after\_arcface}")

Модуль utils

Модуль utils.my\_arcface

Файл main.py

import cv2

import numpy as np

from arcface.lib import l2\_norm

def calculate\_embeddings(img, model):

embeds = l2\_norm(model(img))

return embeds

def get\_distance(embeds1, embeds2):

diff = np.subtract(embeds1, embeds2)

distance = np.sum(np.square(diff))

return distance

def convert\_image(img, input\_size):

img = cv2.resize(img, (input\_size, input\_size))

img = img.astype(np.float32) / 255.

# img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

if len(img.shape) == 3:

img = np.expand\_dims(img, 0)

return img

def calculate\_distance(path1, path2, input\_size, model):

img1 = cv2.imread(path1)

img2 = cv2.imread(path2)

img1 = convert\_image(img1, input\_size)

img2 = convert\_image(img2, input\_size)

embeddings1 = calculate\_embeddings(img1, model)

embeddings2 = calculate\_embeddings(img2, model)

return get\_distance(embeddings1, embeddings2)

Модуль tensorflow

Файл face\_encoding.py

import argparse

import face\_recognition

def get\_encoding(path):

"""

reads image from a given location and calculates its face encoding

(mage is considered to portrait a single person)

:param path: path to the image file

:return: array with 128 elements, containing a single face encoding

"""

image = face\_recognition.load\_image\_file(path)

return face\_recognition.api.face\_encodings(image)[0]

def calculate\_distance(path0, path1):

"""

reads two images from a given location and calculates distance between their face encodings

(each image is considered to portrait a single person)

:param path0: path to the first image file

:param path1:path to the second file

:return: euclidean distance between image encodings

"""

enc0 = get\_encoding(path0)

enc1 = get\_encoding(path1)

return face\_recognition.face\_distance([enc0], enc1)[0]

if \_\_name\_\_ == "\_\_main\_\_":

parser = argparse.ArgumentParser(description="face encoding script")

parser.add\_argument("-p0", dest="path0", required=True, type=str, help="path to the first image")

parser.add\_argument("-p1", dest="path1", required=True, type=str, help="path to the second image")

args = parser.parse\_args()

print('calculating euclidean distance between two face encodings:')

print(f'first: {args.path0}\nsecond: {args.path1}\nresult: {calculate\_distance(args.path0, args.path1)}')

Файл style\_transfer.py

import os

import argparse

import numpy as np

import tensorflow as tf

import tensorflow\_hub as hub

from skimage.io import imshow, imsave

from matplotlib import pyplot as plt

MODEL\_URL = 'https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/2'

os.environ['TFHUB\_MODEL\_LOAD\_FORMAT'] = 'COMPRESSED'

class TransferModel:

"""

a sample class to utilize tf hub style transfer model

(transfers portraits to sketches quite okay,

to perform transfer from sketches to portraits we should find something better)

"""

def \_\_init\_\_(self, url=MODEL\_URL):

"""

initializes a model via tf hub url

:param url: url to tensorflow hub model

"""

self.model = hub.load(url)

print(url)

def transfer\_style(self, content\_img, style\_img):

"""

transfers a style from one image to another

(images should be passed as tf tensors)

:param content\_img: image to be processed

:param style\_img: image which style to be transferred

:return: input image with transferred style

"""

res = self.model(tf.constant(content\_img), tf.constant(style\_img))[0]

return self.\_\_tensor\_to\_image(res)

def process\_image(self, content\_path, style\_path, result\_path, should\_display=False):

"""

performs style transfer and displays the results if needed

:param content\_path: path to image to be processed

:param style\_path: path to image which style to be transferred

:param result\_path: path to save results to

:param should\_display: indicates whether images should be displayed

:return: window with three images if prompted

"""

content\_img = self.\_\_load\_image(content\_path)

style\_img = self.\_\_load\_image(style\_path)

result\_img = self.transfer\_style(content\_img, style\_img)

imsave(result\_path, result\_img)

if should\_display:

fig = plt.figure(figsize=(30, 5))

fig.add\_subplot(1, 3, 1)

imshow(self.\_\_tensor\_to\_image(content\_img))

fig.add\_subplot(1, 3, 2)

imshow(self.\_\_tensor\_to\_image(style\_img))

fig.add\_subplot(1, 3, 3)

imshow(result\_img)

plt.show()

@staticmethod

def \_\_tensor\_to\_image(tensor):

"""

converts tf tensor to RGB uint8 image

:param tensor: input tensor

:return: numpy array with RGB uint8 image

"""

tensor = tensor \* 255

tensor = np.array(tensor, dtype=np.uint8)

if np.ndim(tensor) > 3:

assert tensor.shape[0] == 1

tensor = tensor[0]

return tensor

@staticmethod

def \_\_load\_image(path):

"""

reads image from given path

:param path: path to image

:return: tf tensor

"""

max\_dim = 512

img = tf.io.read\_file(path)

img = tf.image.decode\_image(img, channels=3)

img = tf.image.convert\_image\_dtype(img, tf.float32)

shape = tf.cast(tf.shape(img)[:-1], tf.float32)

long\_dim = max(shape)

scale = max\_dim / long\_dim

new\_shape = tf.cast(shape \* scale, tf.int32)

img = tf.image.resize(img, new\_shape)

img = img[tf.newaxis, :]

return img

if \_\_name\_\_ == "\_\_main\_\_":

parser = argparse.ArgumentParser(description="style transfer script")

parser.add\_argument("-url", dest="url", default=None, type=str, help="url to tensorflow hub model")

parser.add\_argument("-ct", dest="content\_path", required=True, type=str, help="path to content image")

parser.add\_argument("-st", dest="style\_path", required=True, type=str, help="path to style image")

parser.add\_argument("-rs", dest="result\_path", required=True, type=str, help="path to save the result image to")

parser.add\_argument("-d", dest="display", default=True, type=bool, help="prompt to display results")

args = parser.parse\_args()

if args.url:

tm = TransferModel(args.url)

else:

tm = TransferModel()

tm.process\_image(args.content\_path, args.style\_path, args.result\_path, args.display)

**ОТЗЫВ О ПРОХОЖДЕНИИ ПРАКТИКИ**

|  |  |
| --- | --- |
| Вид практики | производственная практика |
|  | (учебная, производственная, преддипломная) |

|  |  |
| --- | --- |
| Тип практики | научно-исследовательская работа |
|  | (учебная, производственная, преддипломная) |

Сроки прохождения практики: с 08.02.2022 г. по 10.06.2022 г.

по направлению подготовки 10.05.03 Информационная безопасность автоматизированных систем

(уровень академического специалитета)

направленность (профиль) «Обеспечение информационной безопасности распределенных информационных систем»

студентом группы № 6412-100503DВ.А. Родиным

|  |  |  |
| --- | --- | --- |
| № п/п | Критерии оценки | Оценка  (по 5-балльной шкале) |
| 1. | Общая систематичность и ответственность работы в ходе практики |  |
| 2. | Достижение планируемых результатов практики |  |
| 3. | Корректность в сборе, анализе и интерпретации представляемых данных |  |
| 4. | Степень личного участия и самостоятельности практиканта в представляемом отчете о практике |  |
| 5. | Качество оформления отчетной документации |  |
|  | **ИТОГОВАЯ ОЦЕНКА[[1]](#footnote-1)\*** |  |

|  |  |  |
| --- | --- | --- |
| Руководитель практики от |  |  |
| университета, ассистент кафедры ГИиИБ |  | Максимов  А.И. |
|  | (подпись) |  |

1. \* Итоговая оценка выставляется как средняя арифметическая оценок по пяти критериям оценки [↑](#footnote-ref-1)