



Институт
интеллектуальных кибернетических систем

Кафедра кибернетики (№ 22)

Направление подготовки 09.03.04 Программная инженерия

Пояснительная записка

к учебно-исследовательской работе студента на тему:

Разработка метода геолокации по серии
изображений на основе глубокого обучения

Группа

Б14-506

Студент

(подпись)

Шедько А.Ю.

(ФИО)

Руководитель

(0-15 баллов)

(подпись)

Трофимов А.Г.

(ФИО)

-

Научный консультант

(0-15 баллов)

(подпись)

(ФИО)

Оценка руководителя _____

Оценка комиссии _____

Члены комиссии



**Факультет Кибернетики и Информационной
безопасности**

Кафедра кибернетики (№ 22)

Направление подготовки 09.04.04 Программная инженерия

Пояснительная записка

к ВКР магистра на тему:

Группа

Студент

(подпись)

(ФИО)

Руководитель

(подпись)

(ФИО)

Научный консультант

(подпись)

(ФИО)

Оценка руководителя _____

Оценка комиссии _____



Институт интеллектуальных кибернетических систем
КАФЕДРА КИБЕРНЕТИКИ

Задание на УИР

Студенту гр. Б14-506
(группа)

Шедько Андрею Юрьевичу
(фио)

ТЕМА УИР

Разработка метода геолокации по серии изображений на основе глубокого обучения

ЗАДАНИЕ

№ п/п	Содержание работы	Форма отчетности	Срок исполнения	Отметка о выполнении Дата, подпись
1.	Аналитическая часть			
1.1.	Обзор методов геолокации по изображениям	Текст ПЗ	18.03.2018	
1.2.	Изучение и сравнительный анализ алгоритмов глубокого обучения с целью выбора подхода к задаче	Текст ПЗ	18.03.2018	
1.3.	Анализ алгоритмов пространственного разбиения поверхности земли для решения задачи классификации	Текст ПЗ	18.03.2018	
1.4.	Анализ существующих решений задачи геолокации по изображениям.	Текст ПЗ	18.03.2018	
1.5.	Анализ возможностей применения подхода transfer learning к проблеме геолокации с помощью глубокого обучения	Текст ПЗ	18.03.2018	
2.	Теоретическая часть			
2.1.	Формальная постановка задачи геолокации по серии изображений	Текст ПЗ	18.03.2018	
2.2.	Выбор/разработка методов оценки точности работы алгоритмов геолокации	Метод	18.03.2018	
2.3.	Модификация существующих решений в области для работы с серией изображений	Алгоритм	18.03.2018	
2.4.	Разработка метода геолокации по серии изображений используя выбранные/разработанные выше алгоритмы/методы	Метод	18.03.2018	
3.	Инженерная часть			
3.1.	Разработать архитектуру для системы (с учетом требований к области применения)	Схемы, Диаграммы	25.03.2018	
3.2.	Проектирование системы геолокации по серии изображений	Схемы, Диаграммы	25.03.2018	
3.3.	Результаты проектирования оформить с помощью диаграмм, схем, описаний. При проектировании использовать язык UML	Схемы, Диаграммы	25.03.2018	
4.	Технологическая и практическая часть			
4.1.	Реализовать разработанные алгоритмы	Исполняемые файлы, исходный текст	25.03.2018	

4.2.	Протестировать систему с помощью сравнения с аналогами. Разработать тестовые примеры для подтверждения исполнения требований.	Исполняемые файлы, исходные тексты тестов и тестовых примеров	Практика	
4.3.	Реализация должна показывать результат лучше чем аналоги на территории РФ	Графики, Таблицы	Практика	
4.4.	Ожидаемым результатом является программное обеспечение позволяющее осуществлять распознавание континента (2500 km), страны(750 km), города (25 km), где сделано фото	Графики, Таблицы, Исполняемые файлы	Практика	
4.5.	При реализации использовать ЯП python и библиотеки keras, tensorflow	Код программы	25.03.2018	
5.	Оформление пояснительной записки (ПЗ) и иллюстративного материала для доклада.	Текст ПЗ, презентация	25.03.2018	

ЛИТЕРАТУРА

1.	Weyand T., Kostrikov I., Philbin J. Planet-photo geolocation with convolutional neural networks //European Conference on Computer Vision. – Springer, Cham, 2016. – С. 37-55.
2.	Babenko A. et al. Neural codes for image retrieval //European conference on computer vision. – Springer, Cham, 2014. – С. 584-599.
3.	Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks //Advances in neural information processing systems. – 2012. – С. 1097-1105.
4.	Hays J., Efros A. A. IM2GPS: estimating geographic information from a single image //Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. – IEEE, 2008. – С. 1-8.
5.	Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. М.// ДМК, 2007
6.	Hastie, Trevor, Tibshirani, Robert and Friedman, Jerome. The elements of statistical learning: data mining, inference and prediction – 2 edition – Springer, 2009.
7.	Hays J., Efros A. A. Large-scale image geolocalization //Multimodal Location Estimation of Videos and Images. – Springer, Cham, 2015. – С. 41-62.
8.	Николенко С., Кадури А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей // Питер Спб – 2018 – С. 480, ISBN: 9785496025362
9.	
10.	

Дата выдачи задания:

Руководитель

Трофимов А.Г.

(ФИО)

« ____ » февраля 2018г.

Студент

Шедько А.Ю.

(ФИО)

Реферат

Пояснительная записка содержит XXX страниц (из них XX страниц приложений). Количество использованных источников – XX. Количество приложений – X.

Ключевые слова:

Целью данной работы является ...

В первой главе проводится обзор и анализ ...

Во второй главе описываются использованные и разработанные/модифицированные методы-/модели/алгоритмы

В третьей главе приводится описание программной реализации и экспериментальной проверки

В приложении ?? описаны основные требования к форматированию пояснительных записок к дипломам и (магистерским) диссертациям.

В приложении ?? представлена общая структура пояснительной записки.

В приложении ?? приведены некоторые дополнительные комментарии к использованию данного шаблона.

Содержание

Введение	8
1 Анализ проблематики задачи геолокации по изображениям	9
1.1 Обзор методов геолокации по изображениям	9
1.2 Изучение и сравнительный анализ алгоритмов глубокого обучения с целью выбора подхода к задаче	9
1.2.1 Глобальные дескрипторы	9
1.2.2 Локальные дескрипторы	12
1.3 Анализ алгоритмов пространственного разбиения поверхности земли для решения задачи классификации	14
1.4 Существующие решения задачи геолокации по изображениям	14
1.5 Анализ возможностей применения подхода transfer learning к проблеме геолокации с помощью глубокого обучения	14
1.6 Выводы	14
1.7 Постановка задачи дипломной работы/курсового проекта	14
2 Разработка моделей и алгоритмов ...	15
2.1 Формальная постановка задачи геолокации по серии изображений	15
2.2 Выбор/разработка методов оценки точности работы алгоритмов геолокации	15
2.3 Модификация существующих решений в области для работы с серией изображений	15
2.4 Разработка метода геолокации по серии изображений используя выбранные/разра- ботанные выше алгоритмы/методы	15
2.5 Выводы	16
3 Инженерная часть	17
3.1 Использование методики «такой-то» для проектирования программных систем «такого- то типа»	17
3.2 Проектирование системы геолокации по серии изображений	17
3.3 Архитектура подсистемы генерации признаков	17
3.4 Архитектура подсистемы классификации	17

3.5	Проектирование протокола взаимодействия подсистем классификации и генерации признаков	17
3.6	Выводы	17
4	Реализация и экспериментальная проверка ...	18
4.1	Выбор инструментальных средств	18
4.2	Состав и структура реализованного программного обеспечения	18
4.3	Разработка тестовых примеров	19
4.4	Результаты тестирования и примеры работы системы	19
4.5	Сравнение реализованного программного обеспечения с существующими аналогами	19
4.6	Выводы	20
	Заключение	21

Введение

Введение всегда содержит краткую характеристику работы по следующим аспектам:

- актуальность:
 - кто и почему в настоящее время интересуется данной проблематикой (в т.ч. для решения каких задач могут быть полезны исследования в данной области),
 - краткая история вопроса (в формате год-фамилия-что сделал),
 - нерешенные вопросы/проблемы;
- новизна работы (что нового привносится данной работой);
- оригинальная суть исследования;
- содержание по главам (по одному абзацу на главу).

Общий объем введения должен не превышать 1,5 страниц (для ПЗ к УИРам может быть чуть меньше).

1. Анализ проблематики задачи геолокации по изображениям

1.1 Обзор методов геолокации по изображениям

Задача определения места съемки фотографии довольно не проста из-за неоднозначности и недостаточности информации, содержащейся в одном изображении. Например, типичная пляжная сцена (море, солнце, песок, небо...) может быть заснята почти в любой точке земли. Даже достопримечательности не всегда могут служить абсолютными ориентирами: Эйфелева башня может указывать на Париж с Елисейскими полями, а может на Лас-Вегас или на село Париж в Челябинской области. В отсутствие подобных ориентиров люди полагаются на такие признаки как язык дорожных знаков, разметку, окружающую флору; опираясь на знания о внешнем мире для уточнения оценки местоположения. Традиционные системы компьютерного зрения часто не обладают подобными сведениями, полагаясь лишь на то что можно почерпнуть из тестовой выборки.

Для решения задачи геолокации применялись:

- расстояния между изображениями вычисляемые при помощи глобальных дескрипторов изображений. Рассматривалось в 2008 году в работе IM2GPS Хейса и Эфроса из Карнеги Меллон [1]. Рассматривается далее более подробно.
- Дополнение данными спутниковой съемки [2].
- Распознавание ориентиров [3].
- Распознавание формы городского горизонта Skyline2GPS [4]

1.2 Изучение и сравнительный анализ алгоритмов глубокого обучения с целью выбора подхода к задаче

Для задач обработки изображений используется множество различных методов. Перечислим некоторые из них.

1.2.1 Глобальные дескрипторы

Глобальные дескрипторы описывают изображение в целом и представляют его в виде векторов признаков. Обычно каждая точка вносит вклад в значение дескриптора. При поиске по коллекции цветных изображений для человека одной из наиболее значимых характеристик является цвет. К тому же он инвариантен относительно расположения объектов и размера изображения, что упрощает его анализ. Самое распространенное представление цвета - это цветовая гистограмма

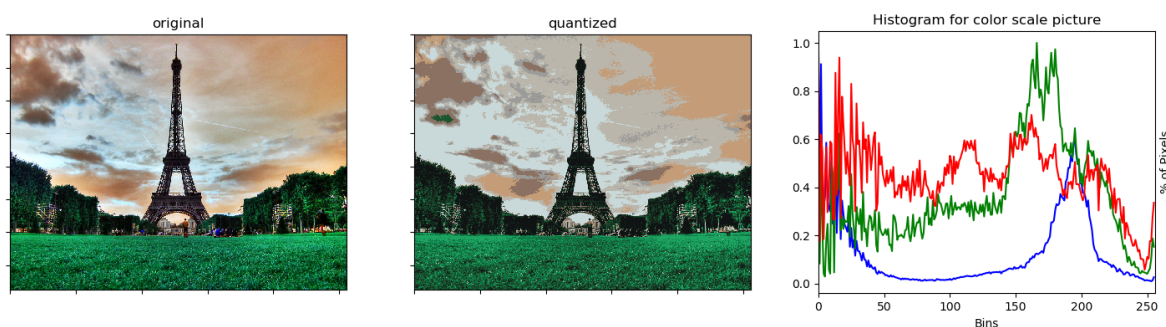


Рис. 1.1 – Гистограмма цветов и квантованная картинка.

Квантование проведено с помощью 10-NN кластеризации

[5] (гистограмма распределения цветов). Подход заключается в том, что цветовое пространство разбивается на промежутки и для каждого промежутка вычисляется доля пикселей из данного промежутка. Процесс разбиения цветового пространства на ограниченное количество цветовых диапазонов называется квантованием. Сложность данного подхода заключается в построении такого разбиения, чтобы цвета из одного диапазона были плохо различимы человеком, а цвета из разных диапазонов, наоборот, различались. Пример гистограммы с квантованием в цветовом пространстве RGB изображен на Рисунке 1.1.

В простейшем случае, и в частности в данном примере, не учитывается пространственное расположение цветов на изображении. Решением является разбиение изображения на фиксированные блоки и вычисление цветовой гистограммы отдельно для каждого блока. Однако в данном решении возникает проблема в подборе размеров блоков. Еще одним решением проблемы учета пространственного расположения цветов является модификация классической цветовой гистограммы HistSP [4]. Главная идея данного метода заключается в том, что для каждого ненулевого элемента гистограммы b вычисляется центр масс пикселей соответствующего цвета, его координаты сохраняются в числе элементов вектора признаков. Пример такой гистограммы изображен на Рисунке 2. Рисунок 2. Цветовая гистограмма HistSP Альтернативная модель представления цвета изображения – это цветовые моменты, предложенные Stricker M., Orengo M. в [5,6]. Авторы рассматривают распределения отдельных цветовых каналов как части трехмерного распределения. Вводится пять фиксированных областей: центральная область в виде эллипса и четыре боковые области (Рисунок 2). Для каждой области вычисляется математическое ожидание по каждому из цветовых каналов и попарные ковариации распределений каналов. Для каждого из пикселей вычисляется его степень принадлежности к области: чем ближе к границе области, тем меньше степень принадлежности к ней. Это значение регулирует вклад цвета соответствующего пикселя в общую оценку распределения цвета области. Рисунок 2. «Нечеткие области» изображения Еще одной

значимой характеристикой изображения является текстура. Она описывает структуру объектов на изображении, и определяется по распределению уровня яркости изображения. Одним из наиболее известных представлений текстурных характеристик изображения являются матрицы смежности (матрицы совместного распределения яркости на изображении - Grey Level Co-occurrence Matrices, GLCM) [7]. Матрица смежности зависит от разницы яркостей в соседних пикселях. По ней можно вычислить значения 7 различных статистических показателей, например, таких как энтропия (степень неоднородности), контраст, показатель однородности, показатель гладкости и др. Еще одними из известных текстурных признаков являются признаки Тамуры [8], которые были выделены с учетом особенностей зрительного восприятия человека. Они включают в себя зернистость, контрастность, направленность, линейность, регулярность и грубость. Из них были определены три наиболее подходящих при решении задачи поиска изображений – это зернистость, контрастность и направленность. Если собрать эти значения в одно изображение, в котором red, green, blue каналы будут заменены на зернистость, контрастность и направленность соответственно, то из полученного изображения можно вычислить 3d текстурную гистограмму. Если будем разбивать на промежутки зернистость, контрастность и направленность, то получим 3d гистограмму, аналогичную цветовой 3d гистограмме на Рисунке 1, только с другими осями координат. Для рассмотрения текстуры изображения в различных масштабах, можно использовать вейвлет-анализ, который заключается в разложении сигнала по базисным функциям. Базисные функции (вейвлеты) строятся на основе порождающего вейвлета с помощью сдвига и масштабирования. Берется исходное изображение и строится первая проекция сигнала (свертка с первой базисной функцией), потом вычисляется разность исходного сигнала с полученным и строится вторая проекция этой разности (свертка со второй базисной функцией), и т.д. Причем, каждая базисная функция является сдвигом предыдущей, растянутой в 2^n раз (n характеризует масштаб). Таким образом, в итоге получаем грубую версию изображения. Такие базисные функции обычно называют фильтрами. Одними из эффективных и используемых фильтров являются фильтры Габора [9], ICA-фильтры (Independent Component Analysis, ICA [10]). ICA-фильтры получены путем анализа обучающего множества изображений. Данные фильтры являются локальными и подобны фильтрам Габора, однако в отличие от них носят естественный характер и отражают основные направления текстуры изображений, по которым они строились. Также проводились исследования, показывающие, что способ построения ICA-фильтров схож с процессом зрительного обучения человека. Еще одним важным признаком для сравнения изображений является форма объектов. Простейшими признаками являются центр тяжести фигур, площадь, направление главной оси и т.д. Существуют и более сложные методы, представляющие фигуры более детально, их можно разделить на два класса:

- внешнее представление, основанное на информации о контуре фигуры – дескрипторы границ, к ним относятся разные виды сигнатур, дескрипторы Фурье, вейвлет-дескрипторы, цепные коды и т.д.;
- внутреннее представление, основанное на информации о фигуре в целом – дескрипторы областей, к ним относятся, например, инварианты моментов и т.д.

Более подробно цветовые признаки, текстурные признаки и признаки формы описаны в

1.2.2 Локальные дескрипторы

Локальные дескрипторы представляют собой вектора признаков, построенные по отдельным фрагментам изображения. То есть они не описывают все изображение, а содержат информацию только о выбранных некоторым способом фрагментах. Самыми известными локальными дескрипторами являются SIFT [11] (Scale Invariant Feature Transform), SURF [12] (Speeded Up Robust Features), PCA-SIFT [13] (PCA – Principal Component Analysis) и т.д. Метод SIFT [11] (Scale Invariant Feature Transform) обнаруживает и описывает локальные особенности изображения. Получаемые с помощью него признаки инвариантны относительно масштаба и поворота, устойчивы к ряду аффинных преобразований, шуму, изменению в освещении. Данный алгоритм можно разделить на две части: определение «точек интереса» (key points, points of interest, salient points) и построение дескрипторов окрестностей данных точек. Существует несколько способов определения точек интереса. Алгоритм, предложенный в рамках SIFT, один из самых известных. Он заключается в использовании пирамиды Гаусса [14], которая строится для изображения. Далее изображения приводятся к одному размеру, и вычисляется их разность (DoG, difference-of-Gaussian images), как показано на Рисунке 3. Причем в качестве кандидатов точек интереса выбираются только те пиксели, которые сильно отличаются от остальных, это делается, например, путем сравнения каждого пикселя изображения с несколькими соседними данного масштаба, с несколькими соответствующими соседями в большем и меньшем масштабе, этот процесс изображен на Рисунке 3. Пиксель выбирается как точка интереса только в том случае, если его яркость является экстремумом. 9 Рисунок 3. Процесс нахождения точек интереса Далее для каждой такой точки интереса вычисляется локальный дескриптор, характеризующий направление градиентов в пикселях некоторой окрестности. Главным минусом SIFT дескрипторов является их высокая размерность и большое количество на изображении. PCA-SIFT [13] (PCA, Principal Component Analysis – анализ главных компонент) дескриптор – одна из вариаций SIFT, в которой уменьшается размерность дескриптора с помощью анализа главных компонент. Это достигается с помощью нахождения пространства собственных векторов, на которое впоследствии проецируются вектора признаков. Альтернативным подходом

является SURF [12] (Speeded Up Robust Features), который в несколько раз быстрее SIFT. В данном подходе для ускорения поиска точек интереса используются интегральные изображения [15]. Значение в каждой точке интегрального изображения вычисляется как сумма значения в данной точке и значений всех точек, которые находятся выше и левее данной. С помощью интегральных изображений за константное время вычисляются так называемые прямоугольные фильтры [16], которые состоят из нескольких прямоугольных областей. SURF в несколько раз быстрее SIFT, менее чувствителен к шуму, к повороту, но чувствителен к изменению освещения или угла, под которым был сделан снимок. Глобальные и локальные дескрипторы обычно используются для решения разных задач. Глобальные дескрипторы в основном применяются для решения задачи общего поиска изображений по содержанию, то есть для поиска по запросу-образцу визуально и семантически похожих изображений. В данном случае важно все изображение в целом, а не отдельные его области, поэтому для решения данной задачи подходят глобальные дескрипторы, характеризующие все изображение. Локальные дескрипторы обычно применяются для решения задачи поиска нечетких дубликатов. Дубликатами считаются 10 изображений одной и той же сцены или объекта, сделанные в разных условиях или разного качества, в частности, изображения одной и той же сцены в разном масштабе или снятые с разных точек, с различием в освещении или с незначительными изменениями фона. При решении данной задачи важно обнаружить сходство отдельных частей изображений, для данных целей обычно применяются локальные дескрипторы, описывающие особенности областей изображений. Все вышеописанные локальные дескрипторы используют общую парадигму, заключающуюся в нахождении точек интереса и построении для каждой из них дескриптора, описывающего ее окрестность. Принципиально другой подход описан в работе [1]. Он основан на свойстве повторяемости (самоподобии) фрагментов на изображении, то есть на наблюдении, что небольшие фрагменты изображения имеют свойство повторяться на нем в том же или другом масштабе. Информация о такой повторяемости в пределах некоторой области изображения формирует так называемую геометрическую разметку. С помощью данной геометрической разметки формируются самоподобные локальные дескрипторы. Причем, даже если изображения имеют разную текстуру, цвет и др., но их геометрические разметки похожи, то дескрипторы тоже будут похожи. Эксперименты, проведенные авторами работы, показали применимость данных дескрипторов для решения задач распознавания объектов и поиска фрагментов в коллекциях изображений и видео без предварительного обучения. Также стоит заметить, что большинство других локальных дескрипторов строятся только для точек интереса и, следовательно, не описывают все изображение в целом. Самоподобные локальные дескрипторы изначально строятся для каждой точки изображения, после чего производится фильтрация, даже после которой самоподобные дескрипторы образуют более плотное множество по сравнению с другими

описанными локальными дескрипторами.

1.3 Анализ алгоритмов пространственного разбиения поверхности земли для решения задачи классификации

...

1.4 Существующие решения задачи геолокации по изображениям

1.5 Анализ возможностей применения подхода transfer learning к проблеме геолокации с помощью глубокого обучения

1.6 Выводы

Подразумевается что разрабатываемая система действует автономно и не обладает дополнительной информацией о мире кроме информации о пространственной плотности объектов тестовой выборки.

Тут пишем выводы по результатам анализа: что и с какой целью было проанализировано, какие выводы из этого сделаны, как они повлияли (должны повлиять) на дальнейший ход работы. Результаты анализа приводятся по пунктно, основные выводы из проделанного анализа. Например:

1. Выполнен сравнительный анализ таких-то формальных систем с точки зрения применимости к решению такой-то задачи. Ни одна из проанализированных напрямую не подходит, поэтому требуется разработать вариацию на основе системы такой-то.
2. Были проанализированы варианты программных архитектур на основе систем. С учетом требований к поддержке больших объемов данных и высоких требований к потенциалу модернизруемости, была выбрана за основу такая-то архитектура.
3. Сравнительный анализ таких-то библиотек показал, что библиотека X проще в использовании, но менее производительна, в то время как библиотека Y обеспечивает высокую производительность, но и требует значительных трудозатрат для использования. В связи с такими-то соображениями были принято решение использовать такую-то библиотеку.

1.7 Постановка задачи дипломной работы/курсового проекта

Это всегда последний пункт. Далее пишется постановка задачи, на основе выданного задания. Это должен быть связный текст в объеме до 1-1,5 страниц. В этом разделе необходимо раскрыть цели и задачи УИРа/диплома.

2. Разработка моделей и алгоритмов ...

В этой главе описываются разработанные/модифицированные модели/методы/ алгоритмы, или/и описывается применение известных стандартных методов. Также, в конце главы обычно приводится общая архитектура программной системы, вытекающая из описанной теории. Приведенные ниже заголовки подразделов так же весьма примерные и сильно зависят от особенностей конкретной работы.

Формулы и их части необходимо набирать в математическом режиме (символ $\$$). Во избежание переноса длинных формул между строками их стоит размещать по центру колонки, например,

$$Sabc = (\lambda xyz.xz(yz))abc = ac(bc),$$

и, если абзац после формулы продолжается, необходимо использовать `\noindent`.

Для набора правил вывода можно использовать пакет `mathpartir.sty`. Правила вывода могут быть вынесены в виде рисунка (см. рис. 2.1).

Для оформления определений, теорем, доказательств и т. п. могут быть использованы соответствующие окружения, например:

Определение 1. (высказывание) Высказыванием называется любое истинное или ложное утверждение.

2.1 Формальная постановка задачи геолокации по серии изображений

2.2 Выбор/разработка методов оценки точности работы алгоритмов геолокации

...

2.3 Модификация существующих решений в области для работы с серией изображений

...

2.4 Разработка метода геолокации по серии изображений используя выбранные/разработанные выше алгоритмы/методы

В ряде случаев, все или некоторые результаты проектирования могут быть представлены во второй главе. Обычно же архитектура описывается в третьей главе.

$$\begin{array}{ccc}
\frac{M \rightarrow M'}{NM \rightarrow NM'} & (\mu) & \frac{M \rightarrow M'}{MN \rightarrow M'N} \quad (\nu) \qquad \frac{M \rightarrow M'}{\lambda x.M \rightarrow \lambda x.M'} \quad (\xi)
\end{array}$$

Рис. 2.1 – Правила редукции

2.5 Выводы

Необходимо перечислить, какие теоретические результаты были получены с указанием степени новизны. Например: «Была разработана такая-то модель. Она представляет собой адаптированную версию модели X , в которой уравнение Z заменено на уравнение Z' ». Еще пример: «Была предложена такая-то архитектура, она отличается от типовой в том-то и том-то. Это позволяет избежать таких-то проблем.». При этом не следует заниматься «высасыванием из пальца»: «Поставленная задача является типовой; для ее решения применены стандартные средства (перечислить, какие).».

3. Инженерная часть

В этой главе описывается, что и как было спроектировано. При необходимости, описывается использованная методика проектирования. Сюда же относится описание внешних и внутренних программных интерфейсов, а также форматы и структуры входных и выходных данных.

3.1 Использование методики «такой-то» для проектирования программных систем «такого-то типа»

...

3.2 Проектирование системы геолокации по серии изображений

...

3.3 Архитектура подсистемы генерации признаков

...

3.4 Архитектура подсистемы классификации

...

3.5 Проектирование протокола взаимодействия подсистем классификации и генерации признаков

...

3.6 Выводы

Следует перечислить, какие инженерные результаты были получены, а именно: какие программные системы, подсистемы или модули были спроектированы. Следует не только назвать полученные архитектуры, но и отметить их отличительные особенности.

4. Реализация и экспериментальная проверка ...

В этой главе описывается, что и как было запрограммировано, отлажено, протестировано, и что в результате получилось. Большинство работ должны содержать приведенные ниже разделы. Но нужно учитывать, что точный состав этой главы, как и других глав, зависит от специфики работы.

Фрагменты программного кода в тексте необходимо выделять при помощи команды `\verb`.

Многострочные листинги должны оформляться при помощи пакета `listingsutf8`. Пример:

```
# let s x y z = x z (y z);;
val s : ('a -> 'b -> 'c) -> ('a -> 'b) -> 'a -> 'c = <fun>
# let k x y = x;;
val k : 'a -> 'b -> 'a = <fun>
# let i = s k k;;
val i : '_a -> '_a = <fun>
```

Листинг 4.1 иллюстрирует использование выносных листингов. Листинг 4.2 показывает пример включения внешнего файла в качестве листинга, в данном случае — выносного.

4.1 Выбор инструментальных средств

В этом разделе обосновывается выбор инструментальных средств; одним из критериев выбора могут быть какие-либо требования к разрабатываемой системе, и если этих требований много, они могут быть выделены в отдельный раздел, или же в приложение. Этот пункт не пишется, если в аналитической главе был раздел, посвященный сравнительному анализу и выбору инструментальных средств.

4.2 Состав и структура реализованного программного обеспечения

Нужно охарактеризовать реализованное ПО: является ли оно настольной программой для Windows, или веб-приложением в форме сайта/веб-сервиса, или модулем/подключаемой библиотекой, или Также нужно перечислить, из чего оно состоит: какие исполняемые файлы и их назначение, конфигурационные файлы, файлы баз данных, требования к программному и аппаратному окружению, и т.п.

Если реализованное приложение достаточно обширно, этот раздел может быть разделен на несколько: один с общим описанием, и по одному на подсистемы самого верхнего уровня.

```
List myList = new List();
Element myElement = new Element();
myList.Append(myElement);
```

Листинг 4.1 – Выносной листинг

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, world!")  
  }  
}
```

Листинг 4.2 – Листинг из файла HelloWorld.scala

4.3 Разработка тестовых примеров

Описываются наиболее характерные тестовые примеры, для прогона на интеграционных тестах. (Да, использование unit-тестирования — это почти всегда хорошо, основное исключение составляют работы, в которых используемый инструментарий по какой-либо причине в принципе исключает такую возможность. Например, что-нибудь вроде Mathematica.)

В этом же разделе могут приводиться и результаты тестирования, включая таблицы и графики. Результаты тестирования могут быть вынесены в отдельный раздел, если много текстового материала и/или использована (не совсем) стандартная методика тестирования (описание которой также нужно привести).

4.4 Результаты тестирования и примеры работы системы

Нужно помнить, что пользователем может быть не только «менеджер» или «человек в белом халате», но и другой программист. Последнее относится, в первую очередь, к реализованным библиотекам. Для «обычных» приложений нередко бывают пользователи нескольких категорий — например, обычный пользователь и администратор. Для каждой категории нужно описать, как выполняются основные функции, предпочтительно, с помощью серии скрин-шотов. Однако считается плохим тоном вставлять длинную вереницу из скрин-шотов: если их много, большую часть нужно выносить в приложение. Для *этого* раздела нормальной является плотность скрин-шотов из расчета: 1 страница скрин-шотов на 1-2 страницы текста.

Замечание. В ПЗ (как УИРа, так и ВКР) следует избегать ситуаций, когда значительную часть основного содержания составляют страницы с иллюстрациями и таблицами, особенно, если такие страницы следуют подряд. В основном тексте следует оставлять лишь самые основные таблицы и рисунки, а остальное — выносить в приложение.

4.5 Сравнение реализованного программного обеспечения с существующими аналогами

В сравнении должно быть отражено, чем полученное ПО выгодно (и невыгодно) отличается от прочих ближайших аналогов. Практика показывает, что аналоги есть всегда. А если нет аналогов,

значит есть частичные решения, которые реализуют какие-то части функционала вашей системы. Тут тоже может быть относительно много таблиц и графиков.

4.6 Выводы

Следует перечислить, какие практические результаты были получены, а именно: какое программное или иное обеспечение было создано. В число результатов могут входить, например, методики тестирования, тестовые примеры (для проверки корректности/оценки характеристик тех или иных алгоритмов) и др. По каждому результату следует сделать вывод, насколько он отличается от известных промышленных аналогов и исследовательских прототипов.

Заключение

В заключении в тезисной форме необходимо отразить результаты работы:

- аналитические (что изучено/проанализировано);
- теоретические;
- инженерные (что спроектировано);
- практические (что реализовано/внедрено).

Примерная формула такая: по каждому указанному пункту приводится по 3-5 результатов, каждый результат излагается в объеме до 5 фраз или предложений.

Также есть смысл привести предполагаемые направления для будущей работы.

Общий объем заключения не должен превышать 1,5 страниц (1 страницы для УИРов).

Список литературы

1. Hays James, Efros Alexei A. IM2GPS: estimating geographic information from a single image // Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on / IEEE. — 2008. — P. 1–8.
2. Lin Tsung-Yi, Belongie Serge, Hays James. Cross-view image geolocalization // Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on / IEEE. — 2013. — P. 891–898.
3. Retrieving landmark and non-landmark images from community photo collections / Yannis Avrithis, Yannis Kalantidis, Giorgos Tolias, Evaggelos Spyrou // Proceedings of the 18th ACM international conference on Multimedia / ACM. — 2010. — P. 153–162.
4. Skyline2gps: Localization in urban canyons using omni-skylines / Srikumar Ramalingam, Sofien Bouaziz, Peter Sturm, Matthew Brand // Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on / IEEE. — 2010. — P. 3816–3823.
5. Color histogram. — 2018. — Jan. — Page Version ID: 821304796. online; accessed: https://en.wikipedia.org/w/index.php?title=Color_histogram&oldid=821304796 (online; accessed: 2018-04-16).