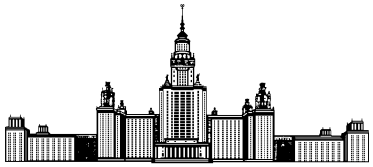
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ДИПЛОМНАЯ РАБОТА СТУДЕНТА 517 ГРУППЫ

«Автоматическая сегментация изображений рукописных документов»

Выполнила:

студентка 5 курса 517 группы

Малышева Екатерина Константиновна

Научный руководитель:

д.т.н., профессор

Местецкий Леонид Моисеевич

Москва, 2014

Содержание

Аннотация

В данной работе предложен алгоритм сегментации рукописных документов на различные классы (в первую очередь разделение на фон, текст и зарисовки). В качестве базы данных использовались отсканированные рукописи А.А. Ахматовой, Б.Л. Пастернака, А.М. Ремизова и М.И. Цветаевой, предоставленные Российским государственным архивом литературы и искусства. В отличие от существующих методов алгоритм устойчив к ориентации текста на изображении и позволяет разделять компоненты на большое количество классов.

1 Введение

1.1 Сегментация и ее применение

Сегментация изображения — это разделение изображения на области, однородные по некоторому критерию. Сегментация, при которой области разбиения не пересекаются, называется тесселяцией. Цель сегментации состоит в упрощении или изменении представления изображения, чтобы его было легче анализировать в дальнейшем. Результатом сегментации является множество сегментов, которые покрывают все изображение. Иначе говоря, каждый пиксель отмечен некоторой меткой некоторого класса.

На сегодняшний день известно большое количество алгоритмов сегментации изображений, использующих разные признаки и подходы. Кроме того, при исследовании сегментации изображений возникает задача оценки качества на некоторой заранее определенной выборке. Соответственно, в решении задачи сегментации необходимо:

1. Создать алгоритм сегментации для существующих данных.

2. Создать критерий для оценки алгоритма качества сегментации.

В данной работе идет исследование обоих пунктов.

Сегментация изображений находит широкое применение в поиске аномалий на медицинских изображениях, в выделении объектов на спутниковых снимках, в системах управления дорожным движением и в подготовительных работах для анализа текста на изображении. В данной работе задача сегментации изображений применяется для получения предварительной разметки рукописных документов.

1.2 Актуальность задачи

На текущий момент огромная часть информации хранится в электронном виде. Поиск и извлечение необходимых знаний происходят намного проще благодаря полуавтоматическим системам навигации по различным корпусам тестов, изображений и видео. Для навигации по рукописным документам нужно выделить объекты, присутствующие на изображении, такие как текст, иллюстрации, печатные вставки.

Таким образом, одной из задач компьютерного зрения является задача распознавания рукописного текста. Этой задаче предшествует сегментация отсканированного документа. Сегментация является важнейшей составляющей алгоритмов распознавания рукописных текстов. Но качество исходного документа часто является очень низким, например, при анализе архивных документов. Происхождение дефектов обусловлено низким качеством бумаги, изношенностью документа, низким разрешением при электронном сканировании.

Российский государственный архив литературы и искусства содержит огромное количество рукописей разных периодов и авторов. Поиск и навигация по архиву происходит вручную, что отнимает большое количество времени. У работников возникают разные сценарии использования архива. Например, найти все зарисовки в черновиках определенного автора. Для такого случая достаточно классификации на текст, изображения и фон. В более сложных случаях класс «текст» может подразделяться на прозу, поэзию, ремарки, исправления и т.д.

1.3 Цели исследования

Целью исследования является создание алгоритма сегментации, который бы давал приемлемое качество для сегментации изображений данного архива. Сегментация в первую очередь должна разделять исходное изображение на фон, текст и зарисовки. Для оценки качества требуется разработать критерий, по которому бы происходило сравнение различных алгоритмов на тестовой выборке.

1.4 Обзор литературы

Множество подходов к решению задачи сегментации текста описано в литературе. В большинстве статей описаны алгоритмы, основанные на предположении о горизонтальной ориентации страницы, а исходными данными служат печатные материалы. Методов работы, ориентированных на распознавание фрагментов с рукописным текстом найти не удалось. Опишем основные методы работы с печатными материалами, описанные в литературе.

В первую группу алгоритмов можно отнести методы, основанные на дроблении страницы на однородные прямоугольные блоки. Далее из блоков выделяются различные признаки и производится кластеризация выбранным алгоритмом. Например, в [1] документ разрезается на однородные прямоугольные блоки фиксированного размера, для каждого блока считается дискретное преобразование Фурье, затем блоки кластеризуются с помощью метода k-средних. На выходе алгоритм возвращает две бинарных маски: для текста и для изображений. В [5] к документам также предъявляется требование о горизонтальном расположении строк. Сначала происходит цветовая коррекция документа, затем разбиение на блоки, вычисление совместного распределения яркости для блока по двум координатам, кластеризация методом k-средних. В [6] на вход алгоритму подаются отсканированные документы, весь текст расположен горизонтально, допустимы изображения любых размеров. Происходит перевод цветного изображения в серое, разбиение на блоки, дискретное вейвлет-преобразование, излечение границ текста, удаление не текстовых регионов. В [7] для блоков вычисляется вейвлет-преобразование и на основе их обучается скрытая марковская модель. Проблема этой группы алгоритмов в строгом требовании к горизонтальной ориентации страниц.

Во вторую группу алгоритмов относятся алгоритмы, основанные на анализе границ бинаризованных компонент. В [2] на вход подается документ, текст в котором расположен под одним и тем же углом. Производится определение ориентации текста с помощью проекции интенсивности изображения. Затем происходит размытие исходного документа, разделение на неоднородные блоки и обход в глубину по границам блоков, на основании которого происходит слияние блоков в различные классы. На выходе имеем бинарное изображение с границами текста в документе. В [3] после пороговой бинаризации изображения происходит вертикальное и горизонтальное сглаживание, поиск границ, объединение разных компонент из эвристических соображений и разделение на компоненты. На выходе алгоритм возвращает 2 бинарных изображения: с текстом и с иллюстрациями.

В основном результатом работы ныне существующих алгоритмов сегментации является бинарная маска, выделяющая текст на изображении. Однако в предложенных алгоритмах не идет исследование распада бинарной маски на отдельные компоненты связности, которые необходимы для сегментации страницы на несколько классов. К тому же исходными

данными в основном являются печатные материалы с одинаковой ориентацией, а не рукописный текст. Эта особенность исходных данных создает сложности из-за разнообразия сегментов, различных шрифтов авторов.

2 Постановка задачи

2.1 Исходные данные

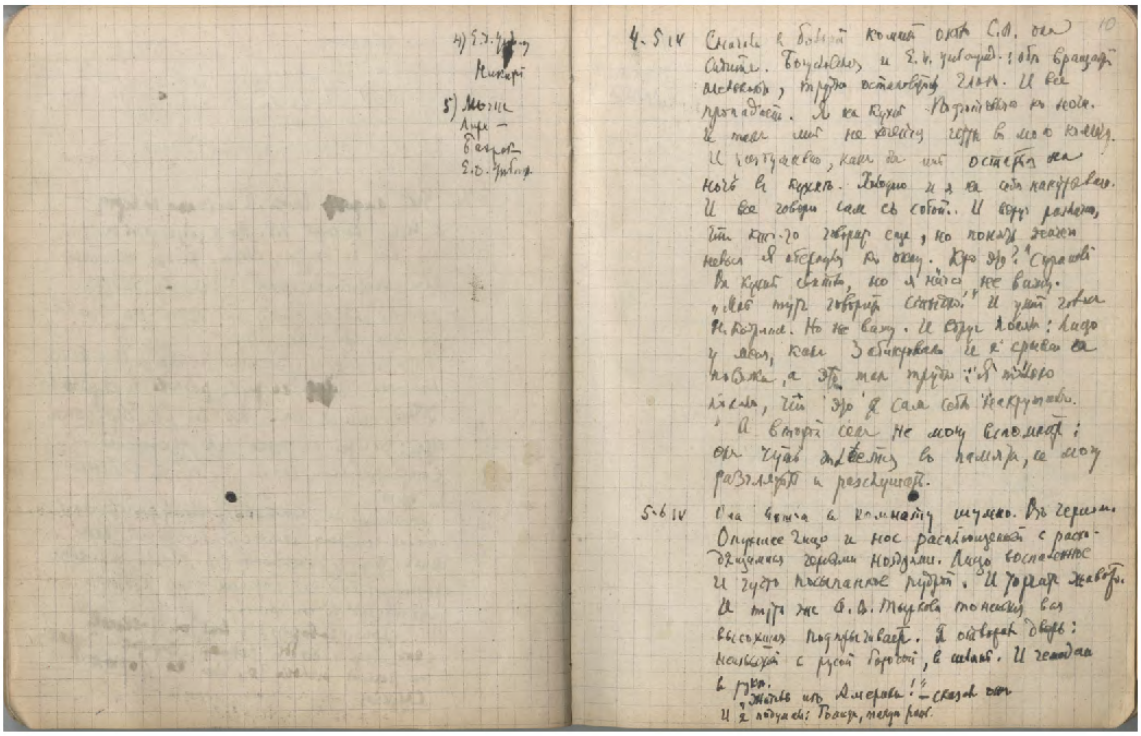


Рис. 1: Пример исходных данных (оригинальное изображение 1975 × 1265 пикселей).

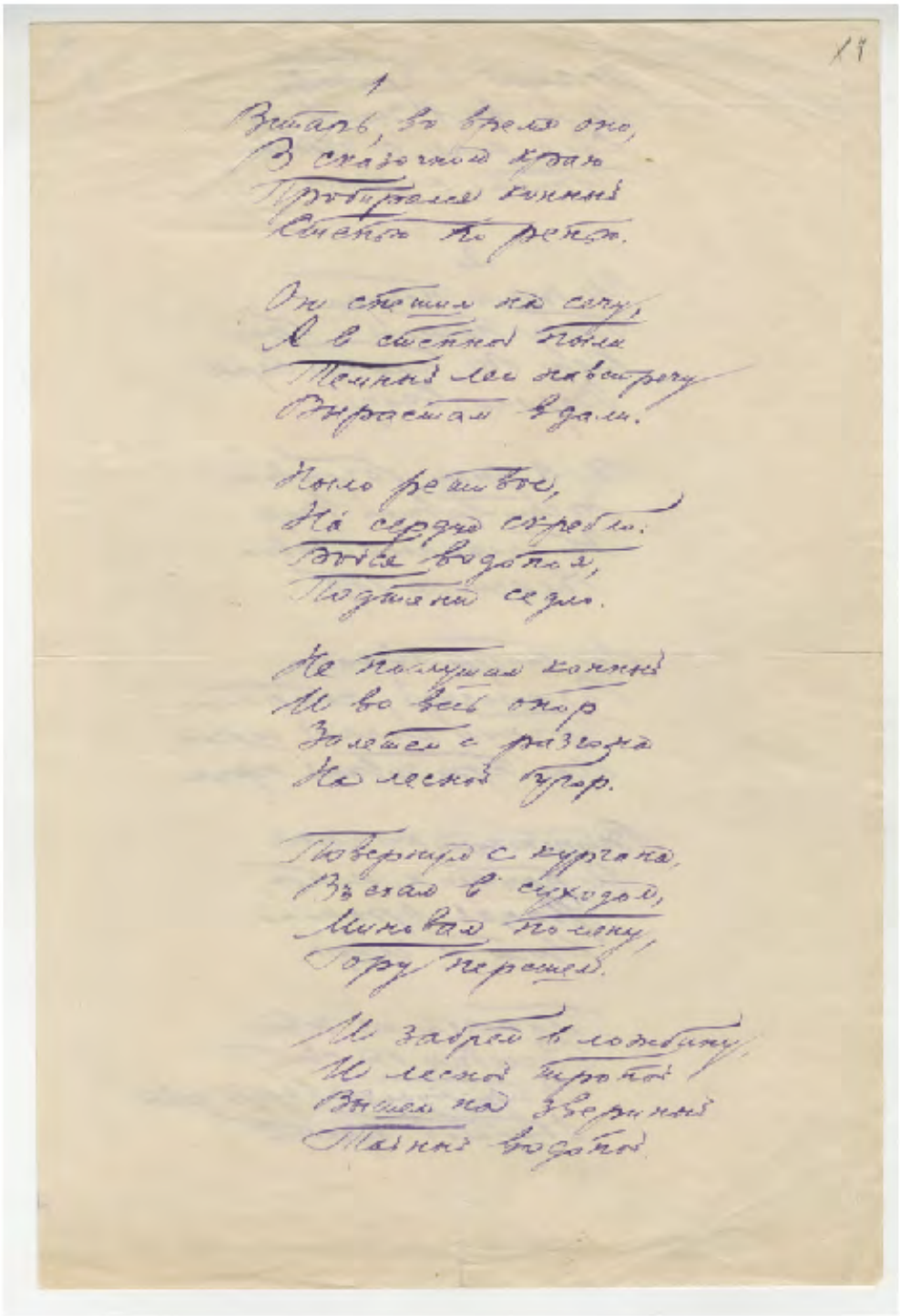
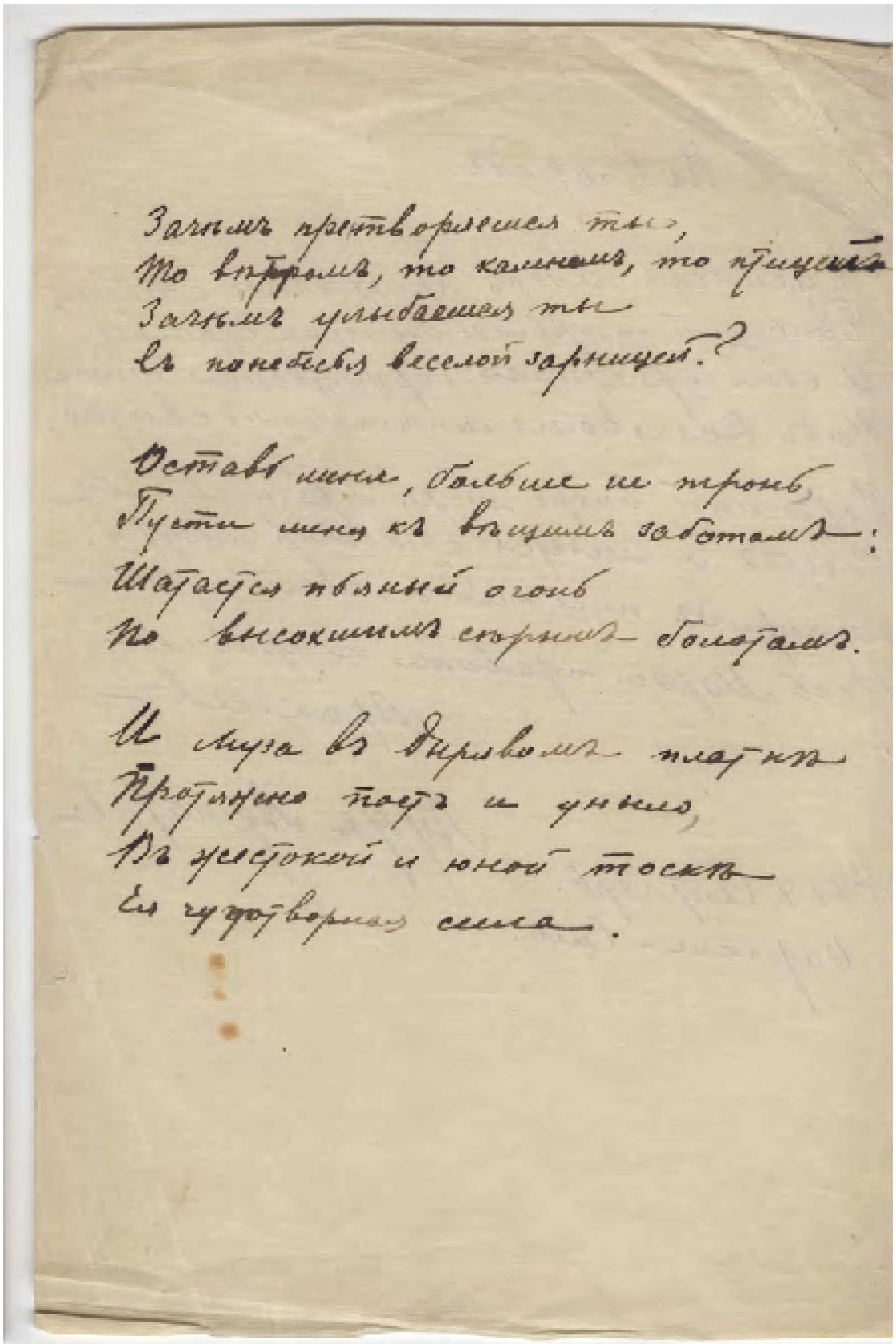


Рис. 2: Пример исходных данных (оригинальное изображение 2583 × 3790 пикселей).


Зачѣмъ претворяешься ты,
То вѣтромъ, то камнемъ, то птицей?
Зачѣмъ улыбаешься ты
Съ понебесья веселой зарницей?

Оставь меня, больше не трогь,
Пусти меня къ вѣщимъ заботамъ:
Шатается пьяный огонь
По высокимъ сѣрымъ болотамъ.

И муза въ дырявомъ платкѣ
Протяжно поетъ и уныло,
Въ жестокой и юной тоскѣ
Ея чудотворная сила.

Рис. 3: Пример исходных данных (оригинальное изображение 1602×2417 пикселей).

Имеются изображения отсканированных рукописей поэтов серебряного века (А.А. Ахматовой, Б.Л. Пастернака, М.И. Цветаевой, А.М. Ремизов, В.Т. Шаламов). Пример исходных данных приведен на Рис. 1, 2, 3.

Отсканированный документ может представлять из себя один лист или разворот тетради. Отнормируем высоту страницы до 1000 пикселей. Посмотрим, какая ширина у архива изображения. Полученная гистограмма изображена на Рис. 4.

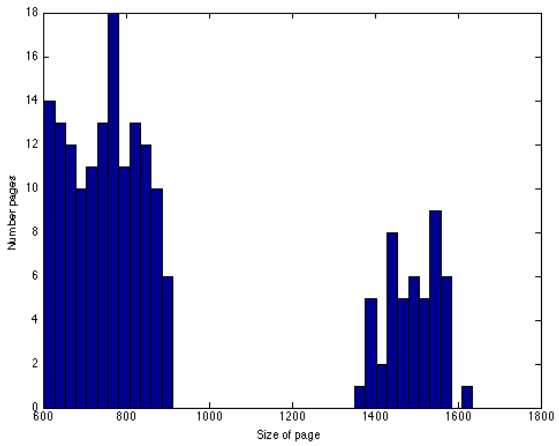


Рис. 4: Распределение размера страниц.

Данное распределение и возможность четкой классификации на один или два листа потребуется при последующей работе алгоритма.

2.2 Формальная постановка задачи

Необходимо создать алгоритм классификации, удовлетворяющий следующим требованиям. На вход алгоритму подается изображение, на выходе получаем размеченное изображение по следующим классам:

1. Заголовок — однострочный текстовый фрагмент, расположенный в верхней части страницы.

2. Стихотворение — многострочный текстовый фрагмент с повторяющейся структурой. Характерен наличием полей вокруг. Как правило, выровнен по левому краю.

3. Прозаический текст — многострочный текстовый фрагмент, структура отсутствует.

4. Ремарка — короткая запись (до 3 строк), расположенная отдельно от стихотворения и прозаического текста, возможна запись по диагонали, на полях и др.

5. Печатные фрагменты — вырезки из газет, билетов, чеков. Отдельный элемент с нерукописным текстом. Пример на Рис. 5.

6. Иллюстрации — зарисовки без текста. Пример на Рис. 6.

7. Нумерация — число, расположенное в правом верхнем углу.

8. Фон — область отсканированного документа, не содержащая значимых объектов.

Результатом работы алгоритма является матрица разметки, где каждый пиксель имеет метку от 1 до 8, в соответствии с классификацией, данной выше.

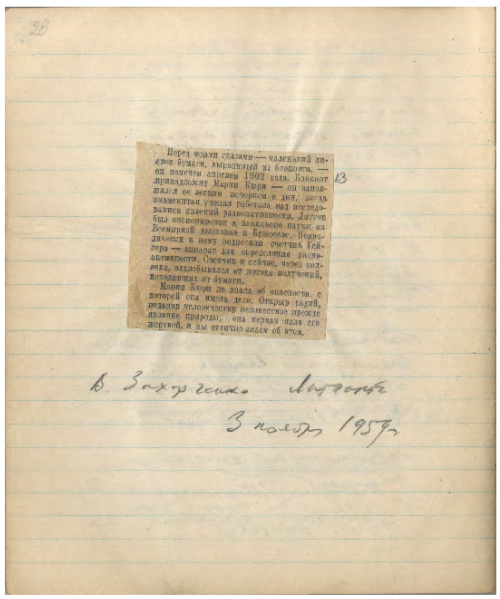


Рис. 5: Пример печатного фрагмента.

Рис. 6: Пример иллюстрации.

Сегментированное изображение задается парой — цветное изображение и разметка на классы.

$$S = (I, L)$$

Цветное изображение I является матрицей размерами $h \times w$. Ее элементы — пиксели, заданные в формате RGB. Разметкой является матрица L размерами $h \times w$. Ее элементы — числа $(1, \ldots, 8)$.

Введем понятие компоненты связности. Зададим граф смежности между пикселями с 8-связной системой соседства. Назовем окрестностью пикселя множество его соседей. Если один пиксель лежит в окрестности другого, то эти пиксели смежны. Пусть имеется множество пикслей V. V называется связным множеством, если для любой пары пикселей p_1, p_2 найдется последовательность пикселей $(p_1, q_1, q_2, \ldots, q_m, p_2)$ в которой пиксели $(q_k, q_{k+1}), \forall k \in \{1, 2, \ldots, m-1\}$ и $(p_1, q_1), (p_2, q_m)$ смежны. Произвольное множество Q распадается однозначным образом на множество не пересекающихся связных подмножеств $C_1, ..., C_n$, которые называются компонентами связности множества Q.

Таким образом, матрица разметок L делится на компоненты связности $C_1, \ldots, C_n$. Каждая компонента представляет из себя пару (B, M): бинарную маску $B_{h \times w}$ и метку класса M. Получаем, что:

$$L = \sum_{i=1}^{n} B_i M_i$$

В итоге сегментированное изображение — это: $S = (I, ((B_1, M_1), \ldots, (B_n, M_n)))$.

Предположим, что существует некоторая истинная сегментация для данного изображения i:

$$S_i = (I, ((B_1, M_1), \ldots, (B_n, M_n))) .$$

Сегментация, полученная неким алгоритмом на изображении i:

$$\widehat{S}_i = \left(I, \left(\left(\widehat{B_1}, \widehat{M_1}\right), \ldots, \left(\widehat{B_m}, \widehat{M_m}\right)\right)\right) .$$

Для оценки качества сегментации необходимо ввести функционал.

Пусть имеется выборка $Images = \{S_1, S_2, \ldots S_W\}$. Тогда для каждого изображения $\forall i \in \{1, 2, \ldots, W\}$ попробуем сопоставить компоненты $C_j \in S_i$ и $\widehat{C_{j'}} \in \widehat{S}_i$.

Пусть центром компоненты $center(C_j)$ называется центр масс данной компоненты.

Пусть площадью компоненты $square(C_j)$ называется количество пикселей в данной компоненте.

Для двух бинарных масок B и B' одинаковых размеров введем меру сходства, как отнормированную симметрическую разность:

$$HD(B, B') = \frac{|xor(B, B')|_{L_1}}{\sqrt{square(B)square(B')}}$$

$\forall j \in \{1, 2, \ldots, m\}$ назовем $\widehat{C_{j'}} = (\widehat{B_{j'}}, \widehat{M_{j'}})$ соответствующей в бинарном смысле для $C_j = (B_j, M_j)$, если:

1. $|center(\widehat{C_{j'}}) - center(C_j)|_{L_2} \leq T_C$
2. $HD(\widehat{B_{j'}}, B_j) < T_S$

Порог T_C (близость центра) и T_S (разность площадей) будут выбраны далее.

После восстановления соответствия в бинарном смысле возможно существуют такие $\widehat{C_{j'}} \in \widehat{S}_i$ и такие $C_j \in S_i$, что для них не нашлось пары. Разобьем эти случаи на 2 множества.

1. Множество компонент $TCWP_i = \{C_j\}$ без пар (true components without pair). То есть, множество компонент из истинной разметки, которые не были найдены алгоритмом.
2. Множество компонент $FCWP_i = \{\widehat{C_{j'}}\}$ без пар (false components without pair). То есть, множество компонент из разметки, полученной от алгоритма, для которых нет пары в оригинальной разметке.

Тогда введем следующий функционал ошибки на $Images = \{S_1, S_2, \ldots S_W\}$ для полученной алгоритмом A бинарной сегментации $\{\widehat{S}_1, \ldots, \widehat{S}_W\}$.

$$Q_b(Images, A) = \frac{1}{W}\sum_{i=1}^{W}\frac{|TCWL_i| + |FCWP_i|}{n_i + m_i}$$

$\forall j \in \{1, 2, \ldots, m\}$ назовем $\widehat{C_{j'}} = (\widehat{B_{j'}}, \widehat{M_{j'}})$ соответствующей в общем смысле для $C_j = (B_j, M_j)$, если они соответствуют друг другу в бианрном смысле и $\widehat{M_{j'}} = M_j$.

После восстановления соответствия в общем смысле возможно существуют такие $\widehat{C_{j'}} \in \widehat{S}_i$ и такие $C_j \in S_i$, что для них не нашлось пары. Разобьем эти случаи на 3 множества.

1. Множество пар $PWWL_i = \{\widehat{C_{j'}}\ , C_j\}$, таких, что $\widehat{C_{j'}}$ и C_j удовлетворяет 1 и 2 условиям, но $\widehat{M_j} \neq M_{j'}$ (pairs with wrong labeling). В данном случае метка компоненты, полученная алгоритмом, отлична от метки истинной сегментации.

2. Множество компонент $TCWP_i = \{C_j\}$ без пар (true components without pair). То есть, множество компонент из истинной разметки, которые не были найдены алгоритмом.

3. Множество компонент $FCWP_i = \{\widehat{C_{j'}}\}$ без пар (false components without pair). То есть, множество компонент из разметки, полученной от алгоритма, для которых нет пары в оригинальной разметке.

Тогда введем следующий функционал ошибки на выборке $Images = \{S_1, S_2, \dots S_W\}$ для полученной алгоритмом A сегментации $\{\widehat{S}_1, \dots, \widehat{S}_W\}$.

$$Q(Images, A) = \frac{1}{W} \sum_{i=1}^{W} \frac{0.5|PWWL_i| + |TCWL_i| + |FCWP_i|}{n_i + m_i}$$

В финальном варианте функционала качества мы хотим штрафовать за неправильную бинарную разметку. За неправильный класс у компоненты мы наказываем, но не так строго.

3 Семантическая сегментация

В работе предлагается метод семантической сегментации. Алгоритм включает следующие этапы. Сначала происходит предобработка изображения линейной коррекцией и применением квантильного фильтра. Далее полученное изображение разбивается на суперпиксели. На следующем этапе для каждого суперпикселя генерируются признаки, на основе которых происходит последующая кластеризация и классификация суперпикселей.

3.1 Предварительная обработка

На первом этапе для последующей сегментации на суперпиксели происходит линейная коррекция изображения и предобработка квантильным фильтром.

Опишем работу квантильного фильтра. Фиксируется 2 параметра: ширина окна и порядковый номер в вариационном ряде отсортированных по яркости соседей пикселя. Вариационный ряд — последовательность значений заданной выборки $x^m = (x_1, \ldots, x_m)$ расположенных в порядке неубывания. Для каждого пикселя имеем его окрестность, размеры которой определяются шириной окна. Элементы окрестности — пиксели. Отсортируем их по яркости. Пикселю присваивается значение соседа, занимающего порядковый номер N в вариационном ряде отсортированных соседей. Пример предварительной обработки изображения показан на Рис. 7 и 9.

Вход: $I_{h\times w}$ - изображение, R - ширина окна, N - порядковый номер пикселя

Выход: $\widehat{I}$ - обработанное изображение;

$(h, w) = size(I);$ // *вычисляем размеры изображения*

для $i = 1, \ldots, h$

для $j = 1, \ldots, w$

$l = \max(i - R, 1);\ r = \min(i + R, h)$

$t = \max(j - R, 1);\ b = \min(j + R, w)$

$neighbourhoods = I(l : r, t : b)$

$neighbourhoods = sort(neighbourhoods)$

$I(i, j) = neighbourhoods(N)$

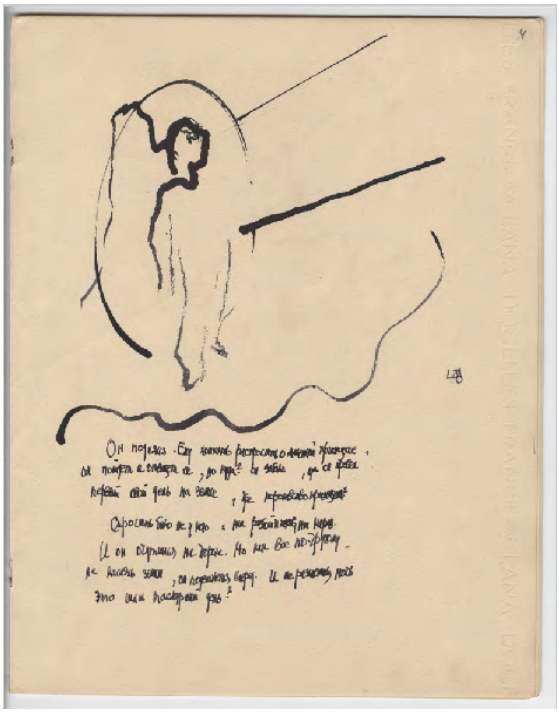


Рис. 7: Исходное изображение,

$h = 1000,\ w = 861$

.

Рис. 8: Изображение после предобработки,

$R = 10,\ N = 20$

.

3.2 Разделение на суперпиксели

Вход: $I_{h \times w}$ - обработанное изображение, NS - количество суперпикселей

Выход: матрица разметки $P_{h \times w}$, элементами которой являются метки p_i, показывающие к какому суперпикселю принадлежит пиксель

На следующем этапе изображение разбивается на так называемые суперпиксели. Суперпиксельная сегментация или предсегментация — это разбиение изображения на фрагменты, которые

1. Компактны, примерно одного размера
2. Границы фрагментов соответствуют границам объектов
3. Достаточно большие, чтобы быть информативными

4. При этом небольшие объекты не должны быть частью сегмента, а должны описываться своим сегментом

В настоящее время для предварительной сегментации на суперпиксели используют следующие группы алгоритмов ([11]):

1. Методы основанные на выделении краев (Probability boundary detection)
2. Эвристические методы (Region growing, Split and Merge, Watershed)
3. Кластеризация (K-Means, Mean shift)
4. Энергетические методы (Snakes, TurboPixels)

В данной работе для разбиения изображения на суперпиксели будет использоваться метод TurboPixels (подробное описание в [10]).

Рис. 9: Исходные данные после предобработки.

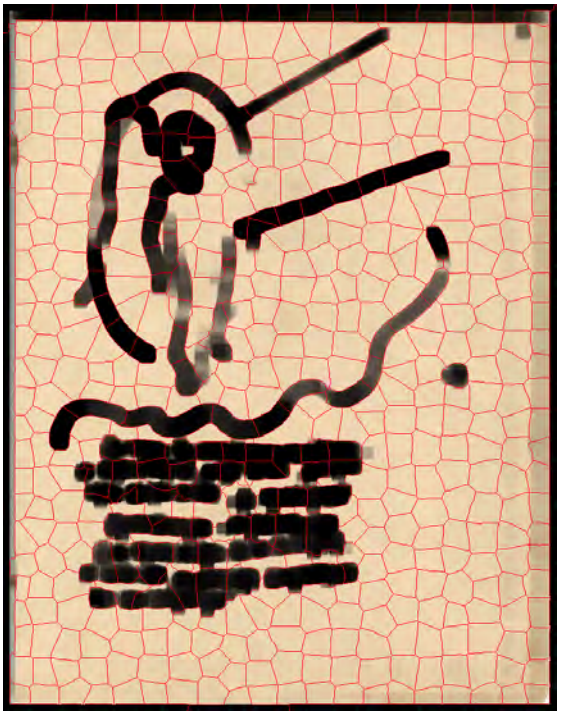


Рис. 10: Сегментация на суперпиксели

Метод TurboPixels достаточно часто используется для предварительной сегментации. Он дает суперпиксели примерно одного размера. В нем используется подход линий уров-

ня для сегментации. Основная идея следующая: для данного изображения выбираются NS начальных точек, равномерно распределенных по изображению. Из них начинается разрастание контура, отвечающего суперпикселю. Скорость движения контура зависит от градиента и близости к предполагаемой границе. Благодаря этому полученные суперпиксели замедляют свой рост изображения и делят его на фрагменты похожего размера.

На выходе работы алгоритма имеем матрицу $P_{h\times w}$, элементами которой являются метки p_i, показывающие к какому суперпикселю принадлежит пиксель. Пример работы алгоритма представлен на Рис.9 и 10.

3.3 Генерация признаков

После разбиения изображения на суперпиксели необходимо преобразовать их в некоторое признаковое описание для дальнейшей кластеризации. На данном этапе на входе имеем матрицу изображения $I_{h\times w}$ и разметку $P_{h\times w}$. NS число суперпикселей на изображении, иначе говоря $max(P) = NS$. В конце данного этапа мы получаем признаковое описание для каждого суперпикселя в виде матрицы $F_{NS\times 1558}$, где строка отвечает за конкретный суперпиксель, а 1588 является мощностью признакового пространства.

Вход: $I_{h\times w}$ - изображение, $P_{h\times w}$ - разметка на суперпиксели.

Выход: $F_{NS\times 1558}$ - признаковое описание. Строка i соответствует суперпикселю p_i.

Опишем структуру признакового пространства. Для каждого суперпикселя необходимо учесть цветовые и текстурные характеристики суперпикселя, а также оценить степень схожести с соседними суперпикселями. В итоге в качетсве признаков были взяты следующие характеристики:

1. Поканальные гистограммы распределений яркости HSV модели.
2. Нормированное количество выделенных пикселей, соответствующих границе после применения фильтра Canny.
3. Гистограмма ориентированных градиентов.
4. Удаленность суперпикселя от края изображения.
5. Степень схожести с соседями.

В качестве цветовой модели была взята модель HSV. Это цветовая модель, в которой координатами цвета являются цветовой тон, насыщенность и яркость. Для суперпикселя производился подсчет количества пикселей с различными HSV характеристиками. Пример поканального разложения представлен на Рис. 11, гистограмма для аналогичных каналов на Рис. 13.

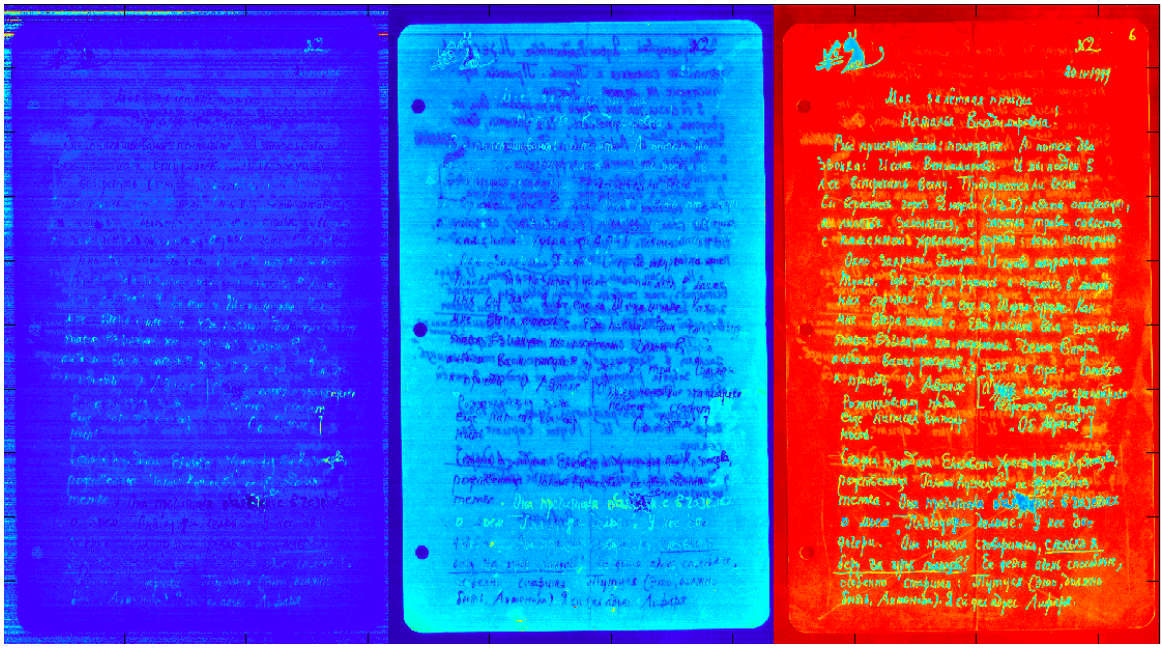


Рис. 11: Поканальное HSV разложение изображения.

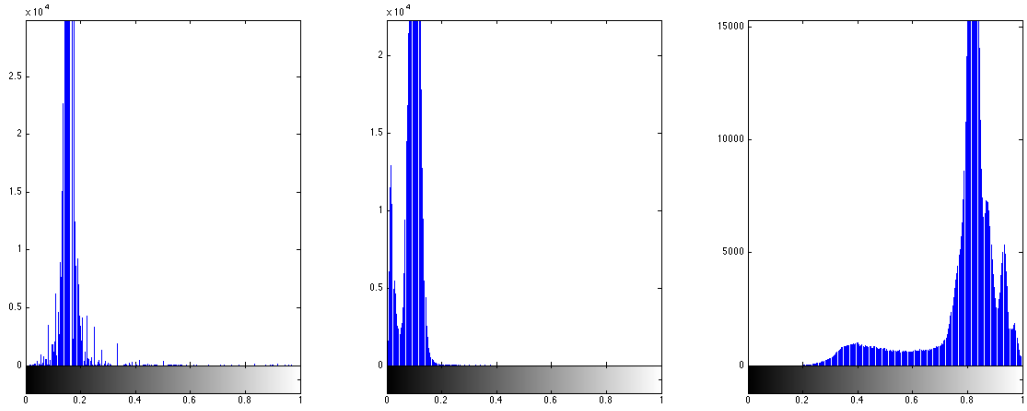


Рис. 12: Поканальная HSV гистограмма изображения.

Для учета текстуры суперпикселя применяется фильтр границ Canny. Алгоритм состоит из пяти отдельных шагов:

1. Сглаживание. Размытие изображения для удаления шума.
2. Поиск градиентов с помощью фильтра Собеля.
3. Подавление не-максимумов. Пикселями границ объявляются пиксели, в которых достигается локальный максимум градиента в направлении вектора градиента. Значение направления должно быть кратно 45 градусам.
4. Двойная пороговая фильтрация (сверху и снизу).
5. Трассировка области неоднозначности. Итоговые границы определяются путём подавления всех краёв, не связанных с определенными (сильными) границами.

На выходе получается бинарное изображение, элементы которого соответствуют границам объектов в суперпикселе (в данном случае элементам рукописных символов). Для перевода результата действия фильтра в признак посчитаем отношение количества белых пикселей к размеру суперпикселя.

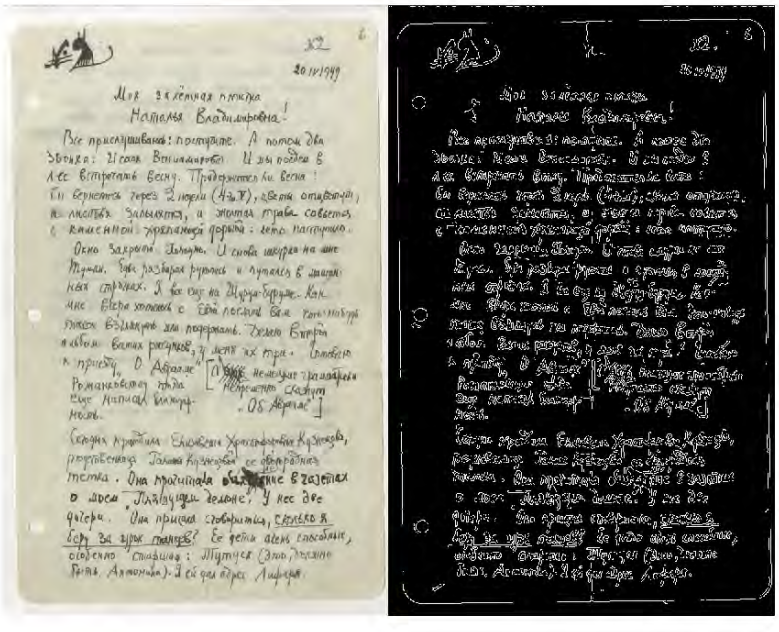


Рис. 13: Оригинальное изображение и бинарная маска после обработки фильтром Canny.

Кроме того для учета текстуры посчитаем дескриптор HOG (histogram of oriented gradients) или гистограмму ориентированных градиентов. Подсчет гистограммы ориентированных градиентов состоит из нескольких шагов.

На первом шаге рассчитываются значения градиентов путем применения одномерной дифференцирующей маски в горизонтальном и вертикальном направлении с помощью свертки яркостной составляющей со следующими фильтрующими ядрами:

$$(-1, 0, 1)$$

$$(-1, 0, 1)^T$$

На следующем шаге происходит подсчет гистограмм для суперпикселя. Каналы гистограммы равномерно распределяются от 0 до 360 градусов (в нашем случае было выбрано 8 секторов: от 0 до 45 градусов и т.д.). При подсчете суммы для канала суперпикселя учитывается абсолютное значение градиента.

Для принятия во внимание яркости и контрастности градиенты нормируются. Дескриптор HOG, таким образом, является вектором компонент нормированных гистограмм суперпикселя. Пример показан на Рис. 14.

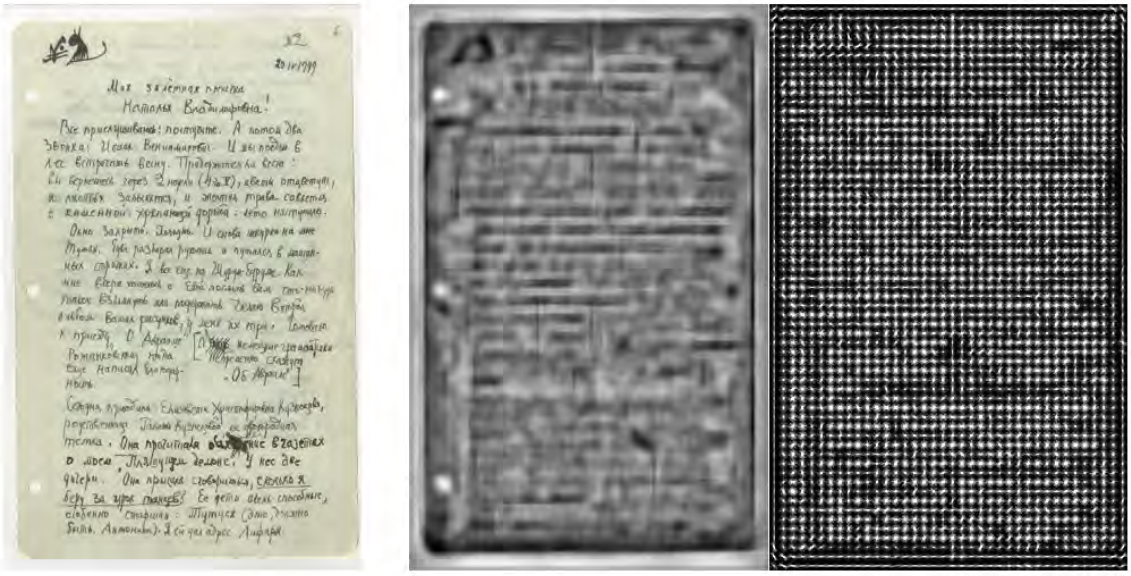


Рис. 14: Визуализация гистограммы ориентированных градиентов.

Еще одним признаком послужили координаты центра масс суперпикселя. По выборке исходных изображений видно, что у отсканированных рукописей есть поля, и по близости к границе можно отсеивать фон.

Итого, унарные признаки для суперпикселя p_i представляют из себя вектор:

$$features(p_i) = [HSV\ hist, canny\ filter, HOG, x, y],$$

размерности:

$$size(features(p_i)) = (3 \times 256, 1, 8, 1, 1) = (779)$$

Для дальнейшей кластеризации важно учесть не только унарные признаки, но еще и окружение суперпикселя, то есть его соседей. Тогда разбиение на суперпиксели может быть представлено в виде графа

$$G = (V, E),$$

где V — множество вершин (суперпиксели), а E — множество ребер. Будем проводить ребро между вершинами, если между соответствующими суперкпикселями есть общая граница в смысле 8-связанности.

Вершина представляет из себя

$$v_i = (p_i, features(p_i)).$$

Путь — последовательность ребер, такая, что конец одного ребра является началом другого ребра.

Длина пути — число ребер в последовательности.

Назовем соседями порядка d для вершины v_i такое множество вершин $NV_i(d) = \{v_{i_1}, v_{i_2}, \dots\}$, для которых существует путь длины d и не существует пути длины меньше, чем d.

Назовем обобщенным соседом d порядка для суперпикселя признаковое описание, полученное как среднее арифметическое от описаний $NV_i(d), NV_i(d-1), \dots NV_i(1)$.

3.4 Кластеризация

Кластерный анализ (кластеризация) — задача разбиения заданной выборки объектов на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кла-

стер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Задача кластеризации относится к широкому классу задач обучения без учителя. Входными данными для задачи кластеризации служит признаковое описание объектов. Каждый объект описывается набором своих характеристик, называемых признаками. Выходными данными задачи кластеризации служат кластеры (непересекающиеся множества).

Формальная постановка задачи кластеризации звучит следующим образом. Пусть X — множество объектов, Y — множество номеров меток кластеров. Имеется выборка объектов $X^m = \{x_1, \ldots, x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по некоторой метрике, а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i.

Сведем задачу объединения суперпикселей в компоненты связности к задаче кластеризации.

Входными данными в задаче кластеризации служат суперпиксели, а их описание — сгенерированные признаки, описанные ранее (матрица F). Выходными данными являются кластеры — непересекающиеся подмножества суперпикселей. То есть у каждого суперпикселя появляется некоторая метка. Для каждой метки формируется бинарная маска, задающая компоненту связности. Кластеризацию суперпикселей попробуем произвести различными алгоритмами. Формально вход и выход данного этапа можно описать следующим образом:

Вход: $F_{max(P)\times 1558}$ - признаковое описание.

Выход: $S_i = (I, (B_1, \ldots, B_n))$ - бинарные маски для различных компонент связности.

В качестве алгоритмов кластеризации были попробованы следующие методы:

1. K-means. Основан на идее минимизации суммарного квадратичного отклонения точек кластеров от центров этих кластеров.

2. Affinity Propagation. Основан на идее «передачи сообщений» между ближайшими точками.

3. Mean Shift. Основан на идее представления данных как смеси гауссиан.

4. Hierarchical clustering. Основан на идее иерархического объединения данных.

5. DBSCAN. Основан на идее разделения объектов на ядровые (с большим количеством объектов в некоторой окрестности) и не ядровые, с дальнейшим объединением в кластеры.

Выбор алгоритма кластеризации произведен в главе «Эксперименты».

3.5 Классификация

Опишем формальную постановку задачи классификации. Имеется некоторое множество описаний объектов X и конечное множество меток классов Y. Существует неизвестная целевая зависимость — отображение $y^* : X \to Y$. Требуется построить алгоритм $a : X \to Y$, способный классифицировать произвольный объект $x \in X$.

Сведем задачу определения класса для компонент связности, полученных на предыдущем шаге, к задаче классификации. Объектами являются $S_i = (I, (B_1, \ldots, B_n))$ - бинарные маски для различных компонент связности. Классами являются: заголовок, стихотворение, прозаический текст, ремарка, печатные фрагменты, иллюстрации, нумерация, фон.

Признаковое описание для каждой компоненты связности составляют:

1. Площадь компоненты S.

2. Координаты центра компоненты (x, y).

3. Эксцентриситет описанного эллипса E.

4. Отношение площади компоненты к площади описанного прямоугольника $S_r = \frac{S}{S_q}$.

5. Поканальные гистограммы распределений яркости HSV.

Проведем разделение классов на некоторые категории и дадим краткое описание каждого класса. На первом этапе разделим все компоненты на рукописные и нерукописные фрагменты, внутри каждого класса перечислим компоненты в порядке возрастания площади.

1. Рукописные фрагменты.

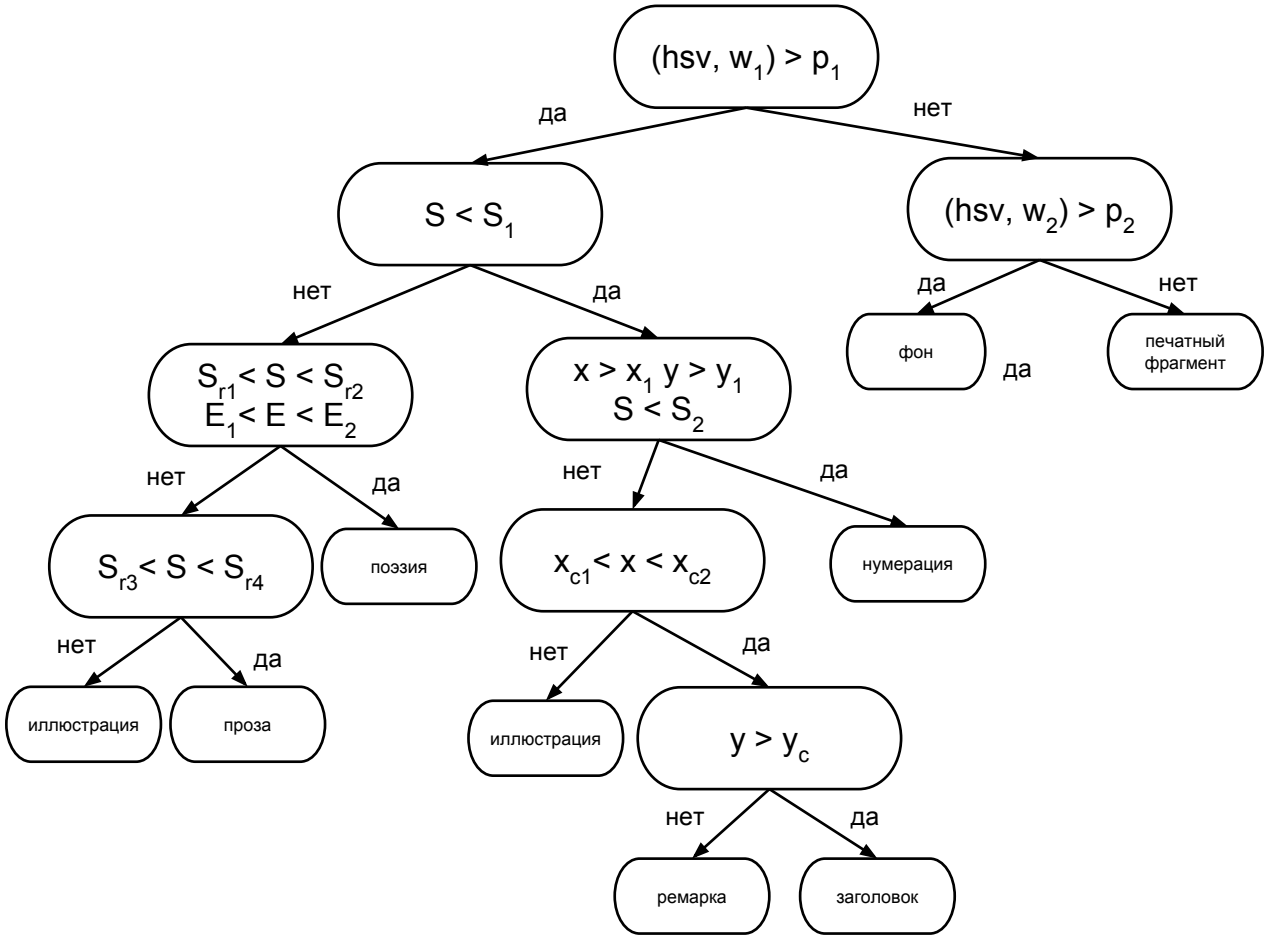


Рис. 15: Дерево решений.

(a) Нумерация — координата по оси x находится ближе к правому краю, координата по оси y ближе к верхнему краю. Выпуклая компонента.

(b) Заголовок — координата по оси x находится ближе в центру изображения, по оси y ближе к верхнему краю. Выпуклая компонента.

(c) Ремарка — координата по оси x находится ближе в центру изображения, по оси y ближе к нижнему краю.

(d) Стихотворение — координата по оси x находится ближе в центру изображения.

(e) Прозаический текст — координата по оси x находится ближе в центру изображения.

(f) Иллюстрации — все остальное.

2. Нерукописные фрагменты.

   (a) Печатные фрагменты — вырезки из газет, билетов, чеков. Отдельный элемент с нерукописным текстом.

   (b) Фон — область отсканированного документа, не содержащая значимых объектов.

По описанным выше признакам и знании о категоризации построим дерево принятия решений для отнесения компоненты к некоторому классу. Полученное дерево решений приведено на Рис. 15.

Приведем некоторые комментарии. В корне дерева разделение происходит на рукописные и не рукописные фрагменты исходя из оценки гистограммы яркости. В рамках нерукописных фрагментов разделение на фон и печатные фрагменты происходит также по сравнению гистограммы яркости. Разделение рукописных фрагментов сначала происходит по площади. К небольшим компонентам относятся: нумерация, ремарка, заголовок. К большим компонентам относятся: проза и поэзия. Компонента «изображение» может быть как небольшой, так и большой компонентой. Внутри небольших компонент разделение на нумерацию, заголовок и ремарку происходит из соображений положения центра. Внутри больших компонент разделение на поэзию и прозу идет на основании эксцентриситета описанного эллипса.

4 Эксперименты

Для оценки качества работы предложенного алгоритма, выбора параметров и алгоритма кластеризации (из списка приведенных выше) была создана тестовая выборка $Images$ с эталонной разметкой. Далее разметка, полученная некоторым алгоритмом A, сравнивалась с эталонной по функционалу качества, введенному ранее.

4.1 Реализация

Для проверки работы предложенного метода был реализован набор алгоритмов, тестирующая система, система для разметки выборки. Разбиение изображения на суперпик-

сели, генерация признаков, тестирующая система, система для разметки выборки были реализованы в системе MatLab. Кластеризация признаков, построение дерева принятия решений в виде набора модулей на языке Python.

4.2 Создание тестовой выборки

В тестовую выборку вошли 36 отсканированных рукописи А.А. Ахматовой, Б.Л. Пастернака, А.М. Ремизова и М.И. Цветаевой и В.Т. Шаламова. Среди них 10 двулистных и 26 однолистных. Для создания эталонной разметки была создана программа-редактор, позволяющая выделять компоненты. На экран выводится изображение. Человек, размечающий выборку, нажатием на курсор выделяет начальную точку. Затем аналогичным образом следующую. Если точка попала в некоторую окрестность уже существующей в компоненте точки, то процесс выделения компоненты прекращается. Запоминаются координаты пути для компонент. Далее происходит или обработка новой компоненты или же переход к обработке следующего изображения. Этапы работы программы представлены на Рис. 16.

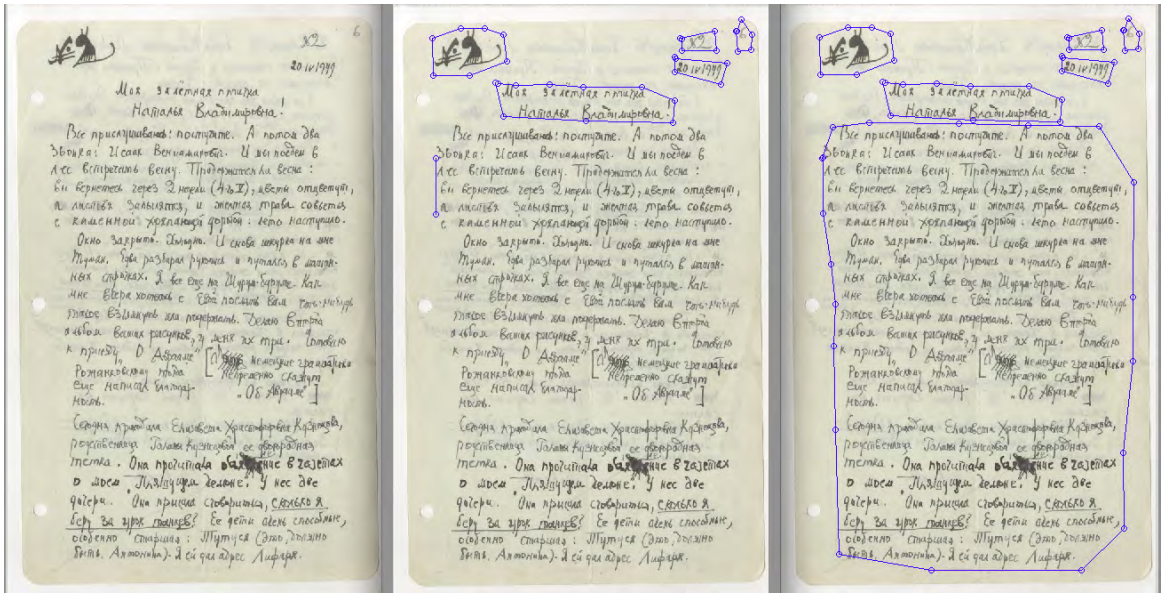


Рис. 16: Разметка изображения на компоненты, работа программы.

В разметке принимали участие 17 человек, каждая картинка была в среднем размечена 7 людьми. На выходе из программы мы получаем координаты компоненты связности, которые переводим в бинарную маску для изображения.

Для полученных бинарных масок от различных людей проведем усреднение. Результат усреднения представлен на Рис. 17.

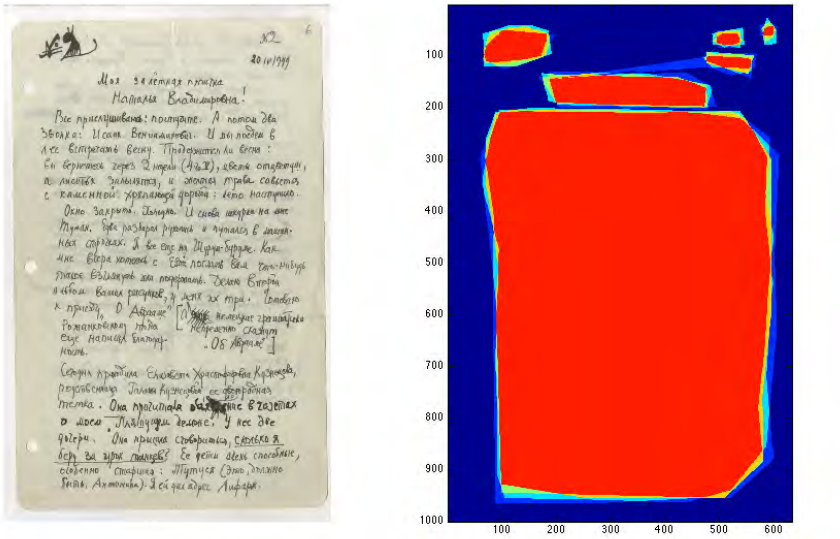


Рис. 17: Усреднение по различным разметкам.

4.3 Результаты эксперимента

Целью экспериментов являлась проверка работоспособности предложенного алгоритма выделения компонент связности и их классификации. В первую очередь оценим качество работы алгоритма выделения компонент.

Опишем параметры запуска алгоритма. На этапе предобработки данных были использованы следующие параметры: квантильный фильтр запускался с радиусом действия $R = 10$ пикселей, соответственно его окрестность в общем (не краевом) случае составляли 100 соседних пикселей. Далее пиксели сортировались по яркости и из выборки брался пиксель под номером $N = 20$. Далее происходило разбиение на суперпикслеи с параметром $NS = 500$. Далее извлекались признаки, согласна данному выше описанию. Алгорит-

мы классификации применялись к тестовой выборке с различной глубиной обобщенного соседа. Признаковое описание суперпикселя составляют его унарные признаки и его обобщенный сосед d порядка.

Проверим работу различных алгоритмов кластеризации компонент для полученной бинарной сегментации в смысле бинарного функционала качества:

$$Q_b(Images, A) = \frac{1}{W}\sum_{i=1}^{W}\frac{|TCWL_i| + |FCWP_i|}{n_i + m_i}.$$

Поясним работу данного функционала на модельном примере.

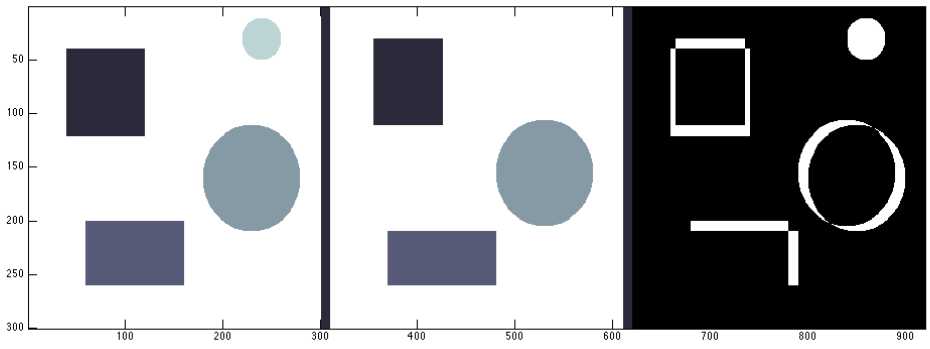


Рис. 18: Пример для оценки качества функционалом. Истинная разметка, ошибочная разметка, результат их симметрической разницы.

На Рис. 18 приведена истинная разметка, ошибочная разметка и результат симметрической разницы для данных изображений. Исходный размер изображения 300×300. При пороге на близость центра $T_C < 20$ и пороге на нормированную площадь симметрической разницы $T_S < 0.2$ для каждой компоненты, за исключение одной, находится соответствующая. Одна компонента осталась без соответствия, поэтому на данном примере значение функционала ошибки в бинарном смысле будет равно $\frac{1}{3+4} \approx 0.14$. При уменьшении порога $T_S < 0.1$ левый верхний четырехугольник перестает соответствовать аналогичной компоненте на соседнем изображении, остальные компоненты продолжают соответствовать друг другу. При таком пороге функционал ошибки в бинарном случае равен $\frac{3}{3+4} \approx 0.42$.

При оценке тестовой выборки было разрешено отклонение по нормированной площади симметрической разницы $T_S < 0.2$ и разница между центрами $T_C < 40$. Для данных

параметров результаты тестирования приведены в Таблице 1. В ячейках указана точность работы алгоритма в процентах, а именно $100(1 - Q_b(Images, A))$.

Таблица 1: Результаты экспериментов.

Алгоритм	$d = 1$	$d = 2$	$d = 3$	$d = 4$
Affinity Propagation	57,3	59,2	60,2	59,8
Mean Shift	58,1	60,3	61,1	58,2
Hierarchical clustering	58,2	58,3	58,8	58,0
K-means	69,1	69,2	69,3	69,1
DBSCAN	77,3	79,3	78,3	75,3

В качестве финального алгоритма для выделения компонент связанности был выбран DBSCAN с параметром $d = 2$.

На следующем этапе проверим работу алгоритма классификации компонент. На вход мы подаем полученные бинарные маски с предыдущего этапа и сравниваем с эталонной разметкой в смысле общего функционала ошибки:

$$Q(Images, A) = \frac{1}{W} \sum_{i=1}^{W} \frac{0.5|PWWL_i| + |TCWL_i| + |FCWP_i|}{n_i + m_i}.$$

Заметим, что ошибка на общем функционале ошибки имеет значение не менее чем на бинарном функционале ошибки, потому что мощность не найденных компонент остается той же самой, но добавляется ошибка неправильной классификации некоторой компоненты. Для приведенного дерева решений и размеченной тестовой выборки точность работы алгоритма в процентах $100(1 - Q(Images, A))$ составила $77,1$.

Пример работы алгоритма приведен на Рис. 19, 20, 21, 22, 23, 24.

Рис. 19: Исходное изображение.

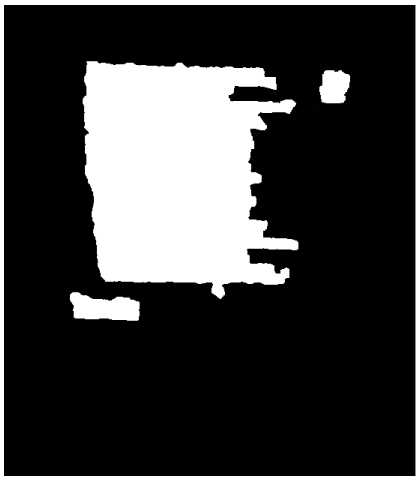


Рис. 20: Бинарная маска.

Рис. 21: Разметка.

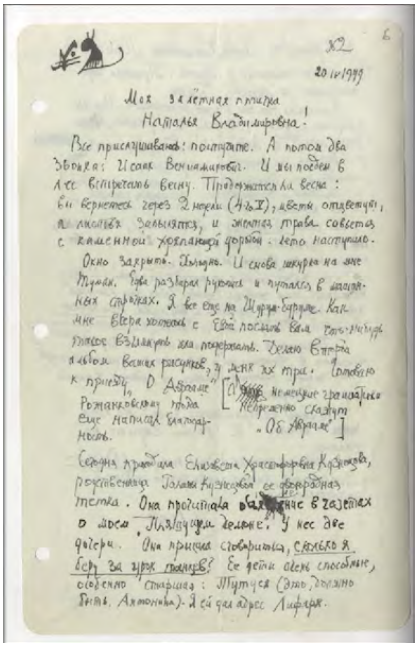


Рис. 22: Исходное изображение.

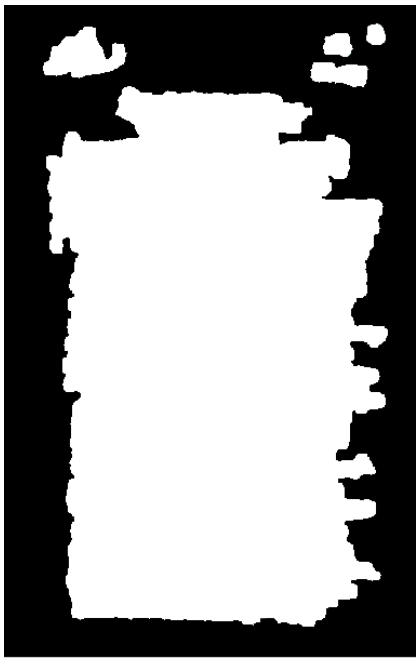


Рис. 23: Бинарная маска.

Рис. 24: Разметка.

5 Заключение

Глобальная задача, которую предстоит решить, заключается в создании полуавтоматической системы навигации по корпусам отсканированных документов. В рамках данной

задачи сделан первый шаг для классификации содержимого листа по классам рукописных материалов на странице. В данной работе был предложен алгоритм по семантической сегментации компонент исходных изображений. Для оценки качества был разработан критерий, по которому бы происходило сравнение различных конфигурация на тестовой выборке. Был реализован набор алгоритмов, создана тестирующая система. Кроме того была собрана эталонная разметка с помощью специальной пользовательской системы для разметки выборки. На введенном функционале качества удалось добиться достойного результата. В рамках данного исследования в дальнейшем планируется улучшить качество поиска за счет добавления признаков и пробы новых методов.

Список литературы

[1] D.Sasirekha, Dr.E.Chandra , *Enhanced Techniques for PDF Image Segmentation and Text Extraction*. 2012.

[2] A. Antonacopoulos , *Page Segmentation Using the Description of the Background*. 1998.

[3] Priti P. Rege, Chanchal A. Chandrakar, *Text-Image Separation in Document Images Using Boundary/Perimeter Detection*. 2012.

[4] C. Strouthopoulos, N. Papamarkovs, A.E. Atsalakis, *Text extraction in complex color documents*. 2001.

[5] M-W Lin, J-R Tapamo, B Ndovie, *A Texture-based Method for Document Segmentation and Classification*. 2006.

[6] Neha Gupta, V .K. Banga, *Image Segmentation for Text Extraction*. 2012.

[7] Sunil Kumar, Rajat Gupta, Nitin Khanna, Student Member, IEEE, Santanu Chaudhury, and Shiv Dutt Joshi, *Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model*. 2007.

[8] А.О. Левашкина, С.В. Поршнев *Исследование супервизорных критериев оценки качества сегментации изображений*. 2008.

[9] Shapiro L. G., Stockman G. C. *Computer Vision*. 2001.

[10] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson *TurboPixels: Fast Superpixels Using Geometric Flows*

[11] Конушин А. С. *Лекции «Сегментация изображений»*