Структурный подход к классификации букв в изображениях

А.Л. Липкина¹, Л.М. Местецкий¹
lipkina96@mail.ru|mestlm@mail.ru

¹Московский государственный университет имени М.В. Ломоносова, Москва, Россия

В работе предлагается математическая модель понятия графемы, основные цели которой — сформулировать строгое определение понятия «графема» и выделить общую структуру изображений одинаковых символов. Построение графемы основано на непрерывном скелетном подходе, который подразумевает построение скелета бинарного изображения символа с последующей его регуляризацией. Также в работе производится применение построенной модели для задачи распознавания текста на цифровом изображении. Для этой цели из модели выделяются признаки, основанные на положениях вершин в модели графемы, и на этих признаках обучается классификатор. Он определяет, к какому классу относится графема, выделенная из бинарного изображения одного символа. Рассматривается также метод обработки входного изображения с текстом для более качественного выделения символов, строк и слов. Проведенные эксперименты показывают работоспособность предлагаемой модели графемы. Алгоритм классификации показывает сравнимые с современными методами распознавания текста результаты.

Ключевые слова: оптическое распознавание символов, цифровое изображение текста, цифровой шрифт, графема, математическая модель, медиальное представление, агрегированный скелетный граф.

A Structural Approach to the Classification of the Letters in the Images

A.L. Lipkina¹, L.M. Mestetskiy¹
lipkina96@mail.ru|mestlm@mail.ru

¹Lomonosov Moscow State University, Moscow, Russia

The paper proposes a mathematical model of the concept of grapheme, the main purposes of which are to formulate a strict definition of the concept of «grapheme» and to highlight the overall structure of images of the same characters. The construction of the grapheme is based on a continuous skeletal approach, which involves the construction of the skeleton of a binary image of the symbol with its subsequent regularization. We also use the constructed model for the problem of text recognition on a digital image. For this purpose, features based on vertex positions in the grapheme model are extracted from the model, and the classifier is trained on these features. It determines which class the grapheme selected from the binary image of one symbol belongs to. We also consider the method of processing the input image with text for better selection of characters, lines and words. The experiments show the performance of the proposed grapheme model. The classification algorithm shows results comparable with modern methods of text recognition.

Keywords: optical character recognition, digital text image, digital font, grapheme, mathematical model, medial representation, aggregated skeleton graph.

1. Введение

Методы оптического распознавания символов (Optical Character Recognition, OCR) активно развиваются с 1950-х годов. Одним из подходов к распознаванию символов вне зависимости от шрифта и размера [8] является метод, основанный на выделении **графемы** [12], [11] из изображения символа. Впервые строгая формализация понятия графемы и применение структуры графемы для классификации букв в изображениях текстов были предложены в [11].

Предлагаемый в [11] метод обладает следующими **недостатками**: наличие отказов от классификации, многоступенчатая структура алгоритма распознавания, сложная система генерации признаков, необходимость большого обучающего корпуса шрифтов для достижения приемлемого качества распознавания.

В данной работе осуществляется попытка улучшения математической модели графемы, упрощение генерации признаков и алгоритма классификации, избавление от отказов классификации, возможность уменьшения обучающего множества шрифтов.

2. Постановка задачи

1. Построение математической модели графемы.
2. Выделение из нее признаков для возможности классификации букв на изображениях текстов.

Основа математической модели графемы — построение *непрерывного представления* скелетного графа бинарного изображения буквы с последующим удалением некоторых рёбер.

Интуитивное понятие графемы

В данном разделе приводятся основные определения, необходимые для введения математической модели графемы. Более подробно они изложены в [11].

Определение 2.1. Графема — графический примитив, имеющий вид геометрического графа и изображающий канонические начертания буквы. Графему можно представить как начертание букв в тонком шрифте. Графемы обладают следующими свойствами:

1. Любые две графемы хорошо *отличимы*.
2. Если два изображения представляются одной графемой, то *различия между изображениями несущественны* (по структуре графемы).

Определение 2.2. Вписанный пустой круг фигуры — круг, целиком лежащий в фигуре и не содержащий ни в каком другом лежащем в фигуре круге фигуры.

Определение 2.3. Скелетное представление фигуры — множество центров всех вписанных пустых кругов.

Скелетное представление фигуры представляет из себя граф $\mathcal{S}$, называемый **скелетом (скелетным графом)** фигуры. Вершинами графа являются центры вписанных пустых кругов, имеющих либо одну, либо три общих точки с границей фигуры, а рёбрами — линии из центров вписанных пустых кругов, касающихся границы ровно в 2 точках. Более подробно см. в [14].

Определение 2.4. Силуэт скелетного графа $\mathcal{V}_S$ — фигура, состоящая из объединения всех вписанных пустых кругов, центры которых лежат в $\mathcal{S}$.

Определение 2.5. Стрижка скелетного графа (с параметром α) — процесс регуляризации скелетного графа $\mathcal{S}$, основанный на удалении несущественных рёбер из скелетного графа. В процессе такого удаления возникает минимальный подграф $\mathcal{S}'$ исходного скелетного графа, для которого выполняется

$$H(\mathcal{V}_S, \mathcal{V}_{S'}) \leq \alpha,$$

где $H(\mathcal{V}_S, \mathcal{V}_{S'})$ — хаусдорфово расстояние [1] между силуэтами скелетных графов $\mathcal{S}$ и $\mathcal{S}'$.

3. Метод решения

Выделение графемы

Пусть на вход подается бинарное изображение одного символа. Тогда алгоритм выделения модели графемы заключается в следующем:

1. Построение непрерывного представления скелетного графа из изображения буквы.
2. Агрегирование скелетного графа.
3. Регуляризация агрегированного графа, то есть удаление шумовых рёбер.

Построение скелетного графа

Построение скелетного графа фигуры подробно описывается в [14]. Основные шаги приведены ниже:

1. Аппроксимация исходной фигуры F многоугольником минимального периметра M .
2. Построение диаграммы Вороного [14] для вершин и сторон многоугольника M .
3. Удаление некоторых отрезков из диаграммы [14].
4. Аппроксимация параболических рёбер диаграммы Вороного прямолинейными рёбрами.

После получения скелетного графа производится операция стрижки с параметром α .

Агрегация скелета

Полученный скелетный граф содержит только следующие типы вершин: *вершины степени 1 (листья)*, *вершины степени 2* и *вершины степени 3 (развилки)*.

Основную информацию о скелетном графе несут листья и развилки, а также виды связей между ними. Чтобы выделить эти связи, производится операция *агрегирования скелетного графа*: «склеивание» в одну цепь всех таких последовательных рёбер, инцидентные вершины которых имеют степень либо 1, либо 2.

После такой «склейки» в качестве вершин остаются только листья и развилки (см. рис. 1).

После агрегации скелетный граф представляет собой гиперграф $\mathcal{S}_{agg,1}$, вершинами которого являются листья и развилки, а рёбрами — выделенные в процессе агрегирования цепочки.

Обозначения и понятия

Все необходимые обозначения приведены в [11]. Кратко приведем их:

1. $\mathcal{R}$ — минимальная по площади прямоугольная рамка с горизонтальными и вертикальными сторонами, ограничивающая бинарное изображение символа. $\mathcal{R}_h$ и $\mathcal{R}_w$ — высота и ширина рамки $\mathcal{R}$.
2. $v_1(e), v_2(e)$ — концевые вершины этого ребра e .
3. $l(e)$ — длина ребра e .
4. Ребро e_{agg} обозначенного гиперграфа $\mathcal{S}_{agg}$ состоит из n последовательных рёбер исходного графа $\mathcal{S}$, попавших в данную цепочку e_{agg} :

$$\{e_{agg}^1, e_{agg}^2, \dots, e_{agg}^n\}.$$

5. $l(e_{agg})$ — длина цепочки e_{agg} .
6. $\deg v$ — степень вершины v .

Определение 3.1. Обозначим отрезок $[v_1(e_{agg}), v_2(e_{agg})]$ через d . Среди всех вершин цепочки e_{agg} находится вершина v_h , наиболее удалённая от отрезка d . По трём точкам $v_1(e_{agg}), v_2(e_{agg}), v_h$ строится окружность. Тогда **аппроксимирующей дугой** называется дуга наименьшей длины, ограниченная точками $v_1(e_{agg}), v_2(e_{agg})$.

Определение 3.2. Пусть ω — окружность, L — некоторая дуга окружности, концами которой являются точки A и B . Пусть O — центр окружности ω . Тогда $\angle BOA$ — **центральный угол** дуги L .

Определение 3.3. Угол кривизны цепочки — центральный угол её аппроксимирующей дуги.

Регуляризация графемы

Наличие или отсутствие засечек не мешает человеку понять, какая именно буква изображена. То есть, модель графемы у буквы в шрифтах с засечками или без должна быть одинаковой. Поэтому следующий этап построения математической модели графемы — удаление из $\mathcal{S}_{agg,1}$ рёбер, являющихся засечками (см. рис. 1). Множество рёбер гиперграфа $\mathcal{S}_{agg,1}$, являющихся засечками, обозначим через $\mathcal{E}_S$.

Рис. 1. 1а: Агрегированный скелетный граф; 2а: Скелет буквы в шрифте с засечками; 2б: Скелет после регуляризации; 2с: Скелет буквы в шрифте без засечек.

Ниже описывается одно из ключевых отличий от метода регуляризации графемы в [11].

Построение множества $\mathcal{E}_S$ происходит в 2 этапа. На первом этапе в множество добавляют те ребра e_{agg} , для которых можно быть уверенным, что они являются засечками. В таком случае ребро e_{agg} попадает в множество $\mathcal{E}_S$, если выполнены все следующие свойства:

- 1) ровно одна из вершин $\{v_1(e_{agg}), v_2(e_{agg})\}$ — листовая, и ровно одна из них является развилкой;
- 2) длина ребра $l(e_{agg})$ не превышает порога $\mathcal{L}(\mathcal{R})$;
- 3) центральный угол 2β аппроксимирующей e_{agg} дуги не меньше некоторого порога $\mathcal{A}$.

На втором этапе происходит расширение множества $\mathcal{E}_S$ по следующему правилу: для всех вершин степени 3, для которых одно из смежных ребер было добавлено в $\mathcal{E}_S$, проделывается:

1. Ищется смежное ребро e'_{agg} , второй конец которого имеет степень 1 и которое не было добавлено в $\mathcal{E}_S$. Если таких ребер несколько, то выбирается кратчайшее из них.
2. Если длина выбранного $l(e'_{agg})$ не превосходит порога $\mathcal{L}(\mathcal{R})$, то ребро e'_{agg} добавляется в $\mathcal{E}_S$.

Идея такого добавления в том, что, как правило, засечки имеют парную структуру, то есть из одной вершины степени 3 выходят 2 ребра-засечки.

Алгоритм удаления шумовых рёбер из $\mathcal{S}_{agg,1}$:

1. Определение множества $\mathcal{E}_S$.
2. Удаление из $\mathcal{S}_{agg,1}$ всех рёбер из $\mathcal{E}_S$.

Так как после удаления могут возникнуть вершины степени 2, необходимо провести повторное агрегирование скелета. Гиперграф, полученный после удаления рёбер и повторной агрегации, обозначим через $\mathcal{S}_{agg,2}$. Этот гиперграф и является предлагаемой **математической моделью графемы**.

Построение признаков из графемы

Второе ключевое отличие от [11] состоит в упрощенном способе генерации признаков из графемы. Идея извлечения признаков состоит в выделении ключевых вершин степени 1 и 3, а также промежуточных вершин степени 2. Вершины степени 2 ищутся следующим способом: для каждого ребра из $\mathcal{S}_{agg,2}$ отмечается середина ребра и она объявляется вершиной степени 2.

Далее используется идея **зонирования** [5, 9]. Для этого рамка изображения $\mathcal{R}$ разбивается на n частей горизонтальными прямыми и m частей — вертикальными. В каждой из получившихся областей отдельно считается количество вершин степени 1, степени 3 и степени 2 (см. рис. 2).

Рис. 2. 2a: графема $\mathcal{S}_{agg,2}$ буквы «К» и разбиение рамки на 9 прямоугольников ($n = m = 3$); 2b: количества листьев в каждом из прямоугольников; 2c: количества развилок; 2d: количества середин рёбер.

Такая генерация признаков производится для N_Z различных масок разбиения на зоны $\mathcal{Z}$. Полученные признаки объединяются в один признаковый вектор, и к ним добавляется признак «число компонент связностей на изображении в рамке $\mathcal{R}$ ».

$$\mathcal{Z} = \{(n_i, m_i)\}_{i=1}^{N_Z}.$$

4. Обработка входного изображения

На вход метод получает цветное изображение X^{RGB} сканированного текста на кириллице. Для возможности выделения текста, входное изображение должно быть бинаризовано и очищено от шумов. Пусть значения пикселей лежат на отрезке $[0, 1]$.

Сначала изображение приводится в серые тона X^{Grey} . Так как фон может быть неравномерно освещен, то для выравнивания освещения из X^{Grey} вычитается фон B^{Grey} . Фон считается при помощи **фильтра Гаусса** [3] с большой дисперсией σ_1 .

$$\begin{cases} B^{Grey} = \text{Gaussian_filter}(X^{Grey}, \sigma_1) \\ f(X^{Grey}) = \frac{X^{Grey}}{B^{Grey}}, \end{cases}$$

где деление является попиксельным.

После этого контраст полученного изображения увеличивается при помощи **линейного растяжения** с квантилями q^{bl}, q^{wh} . Для повышения интенсивности пикселей тёмных оттенков производится нелинейное изменение значения пикселей операцией **гамма-коррекции**. Для улучшения качества изображения за счет фильтрации шумов используется **фильтр Винера** [6]. Далее изображение бинаризуется при помощи метода Оцу [7] и из полученного изображения удаляются объекты меньшей площади s . Получается бинарное изображение X^{BW} .

Выделение символов

На изображении X^{BW} выделяются связные компоненты черных пикселей по системе 4-соседства. В такой системе соседними считаются 2 пикселя, если они граничат по одной из сторон пикселя. Далее, для каждой связной компоненты ищется минимально содержащий её прямоугольник R .

Не все найденные компоненты связности являются буквами, так как на бинарном изображении все еще могут присутствовать посторонние шумы. Для их фильтрации считаются следующие статистики: m_{sq}, m_h, m_w — медианы распределения площадей, высот и широт прямоугольников соответственно.

Пусть $\mathcal{R}$ — текущий прямоугольник, $\mathcal{R}_h, \mathcal{R}_w, \mathcal{R}_{sq}$ — его высота, ширина и площадь соответственно. Шумовым прямоугольником считается хотя бы в одном из следующих случаев:

1. $\mathcal{R}_{sq} \leq \kappa_1 m_{sq}$ и $|\mathcal{R}_h - m_h| > \kappa_2 m_h$.
2. $\mathcal{R}_w \geq \kappa_3 m_w$ или $\mathcal{R}_h \geq \kappa_3 m_h$.

После такой фильтрации из кандидатов на буквы исчезают диакритические знаки (для букв ё и й). Чтобы объединить букву и диакритический знак, производится увеличение высоты прямоугольника:

Если $|\mathcal{R}_h - m_h| < \kappa_4 m_h$, то
высота увеличивается на κ_5 процентов.

После увеличения высоты в рамках каждого прямоугольника заново производится поиск минимального содержащего символ прямоугольника.

Выделение строк

Предполагаем, что текст на изображении представлен в виде горизонтальных (возможно, с небольшими отклонениями) строк с текстом. Пусть $\mathcal{R}_{la,x}, \mathcal{R}_{la,y}$ — левая верхняя x - и y -координаты прямоугольника $\mathcal{R}$.

Строки выделяются по следующему правилу: два символа с ограничивающими их рамками $\mathcal{R}^1$ и $\mathcal{R}^2$ находятся на одной строке тогда и только тогда, когда:

$$|\mathcal{R}_{la,x}^1 - \mathcal{R}_{la,x}^2| < \frac{\kappa_6}{2} (\mathcal{R}_h^1 + \mathcal{R}_h^2)$$

Далее, в рамках одной строки символы упорядочиваются по возрастанию координаты $\mathcal{R}_{la,y}$.

Выделение слов

Для возможности выделения слов считается медиана распределения m_b ширин зазора между рамками букв, лежащими на одной строке. Далее используется следующее правило: два последовательных символа с ограничивающими их рамками $\mathcal{R}^1$ и $\mathcal{R}^2$ разделяются пробелом тогда и только тогда, когда:

$$\mathcal{R}_{ra,y}^2 - \mathcal{R}_{ra,y}^1 > \kappa_7 m_b,$$

где $\mathcal{R}_{ra,y}$ — y -координата верхнего правого угла $\mathcal{R}$.

5. Алгоритм классификации

После обработки входного изображения, выделения символов и разбиения на строки слова производится этап классификации каждого символа. Для этого нужно проделать следующие шаги:

1. Для каждого символа выделить графему.
2. Из каждой модели графему выделить признаки.
3. Применить обученный классификатор.

Данная задача классификации является *многоклассовой*. Пусть в задаче C классов и классификатор a выдает вероятности принадлежности объекта x к классу c_i , $i \in [1, C]$:

$$a(x) = \{\mathbb{P}(x \in c_i)\}_{i=1}^C.$$

Тогда класс для объекта x согласно классификатору a определяется по следующей формуле:

$$class(x) = \arg \max_{c_i} \mathbb{P}(x \in c_i).$$

6. Метрики качества

Этап обучения

Для подбора параметров модели используется метод кросс-валидации. Пусть $(\mathbb{X}_{val}, \mathbb{Y}_{val}) = \{(x_{val}^i, y_{val}^i)\}_{i=1}^{n_{val}}$ — валидационная выборка. Тогда метрика качества Q модели a измеряется при помощи **точности классификации (accuracy_score)**:

$$Q(a, (\mathbb{X}_{val}, \mathbb{Y}_{val})) = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \mathbb{I}[y_{val}^i = class(a(x_{val}^i))].$$

Этап тестирования

На этапе тестирования в качестве исследуемых метрик используются *точность классификации* и **редакторское расстояние (расстояние Левенштейна)** [13] $L(s_1, s_2)$ между двумя строками s_1, s_2 . $L(s_1, s_2)$ считается как минимальное количество замен, вставок и удаления букв из s_1 , необходимое для превращения строки s_1 в строку s_2 . Также считается **относительное расстояние Левенштейна** $L'(s_1, s_2)$, равное:

$$L'(s_1, s_2) = \frac{L(s_1, s_2)}{length(s_2)},$$

где $length(s)$ — длина строки s .

7. Вычислительные эксперименты

Обучающая выборка

1. Выбирается количество шрифтов N_f , в данной работе — до 116 видов, включая различные стили написания (bold, condensed, normal, light и др.), а также с засечками или без (serif/sans) [8].
2. Выбираются размеры шрифта, с которыми будет генерироваться выборка (всего N_s размеров). Предложенный классификатор *может обучаться на изображениях разного размера*.
3. Программно генерируются бинарные изображения букв *без опции сглаживания*. Выборку, сгенерированную таким способом, назовём **оригинальной**.
4. Программно генерируются бинарные изображения букв *без опции сглаживания*, затем они зашумляются при помощи фильтра Гаусса с дисперсией σ_2 , и после этого заново бинаризируются методом Оцу [7].
5. Генерируются программно бинарные изображения букв *с опцией сглаживания*, затем применяется фильтр Гаусса с дисперсией σ_2 , и заново бинаризируются методом Оцу [7]. Выборку, сгенерированную тремя вышеописанными способами (3–5) будем называть **зашумленной**.
6. Каждый раз генерируется по 68 букв из каждого шрифта и размера: 34 строчные и 34 прописные. Буква **Ы** может состоять из 2 графем: **Ь** и **И**. Поэтому к набору из 33 букв добавляется еще одна графема, отвечающая букве **И**.
7. В качестве истинного класса изображения буквы используется *строчное написание буквы*, то есть число классов $C = 34$. Приведение к нижнему регистру обосновано тем, что существует ряд букв (например, **С** и **с**), заглавное и строчное написание которых представляют собой одну графему. В этом случае классификатор может сделать ошибку.
8. Размер выборки составляет $3 \cdot 68 \cdot N_f \cdot N_s$ объектов.

Значения параметров

$$\alpha = \max(0.06 \cdot \mathcal{R}_h, 1), \quad \mathcal{A} = \frac{\pi}{5}, \quad \mathcal{L}(\mathcal{B}) = \frac{2}{7} \max(\mathcal{R}_h, \mathcal{R}_w),$$

$$\kappa_1 = 0.3, \quad \kappa_2 = 0.3, \quad \kappa_3 = 5, \quad \kappa_4 = 0.5$$

$$\kappa_5 = 35, \quad \kappa_6 = 0.7, \quad \kappa_7 = 3,$$

$$\sigma_1 = 50, \quad \sigma_2 = 1, \quad q^{bl} = 0.03, \quad q^{wh} = 0.97, \quad s = 10,$$

$$\mathcal{Z} = \{(3, 3), (2, 2), (2, 3), (3, 2), (4, 4),$$

$$(4, 3), (3, 4), (3, 8), (4, 8), (5, 3), (3, 5)\}.$$

Классификатор a — случайный лес [4].

Базовый алгоритм

В качестве базового алгоритма выбрана свободно распространяемая OCR-система **tesseract** [10]. В её основе — **рекуррентная нейронная сеть LSTM** [2].

Эксперимент 1

Исследуется зависимость качества распознавания от размера и качества обучающей выборки.

Фиксируются количества шрифтов для обучения: **10, 20, 30** размера **50pt**. Для каждого набора шрифтов генерируется 2 обучающие выборки: *оригинальная (original)* и *зашумленная (noisy)*. На каждой из выборок обучается своя модель классификации. Для теста:

1. Генерируются выборки-дублиеры обучающего множества, но в другом размере шрифта (**30pt**).
2. Генерируется выборка из 50 шрифтов размера **30pt**, не участвующих во всех обучающих множествах.
3. Выборки — в *оригинальной* и *зашумленной* формах. Результаты приведены в таблице 2.

Эксперимент 2

В данном эксперименте проводится анализ ошибок обученных выше классификаторов. Исследуется, для каких букв классификаторы чаще всего ошибаются. В таблице 3 представлен топ-3 по убыванию ошибок классификатора в формате:

$p(\text{предсказанная буква} | \text{правильная буква}) =$ процент случаев, в котором классификатор ошибся для данной правильной буквы.

Эксперимент 3

В данном эксперименте исследуется качество предложенного метода в сравнении с базовым алгоритмом **tesseract** на *реальных данных*.

Для тренировочной выборки фиксируется 116 шрифтов. Выборка генерируется в зашумленном варианте. Тестовая выборка состоит из размеченных скриншотов, сканов и фотографий реальных текстов. Её объем — около 40 изображений. Метрики качества — *расстояние Левенштейна L* и *относительное расстояние Левенштейна L'* . В относительном деление происходит на длину текста-ответа без учета знаков пунктуации. Результаты эксперимента приведены в таблице 4. Примеры распознанного текста — на рис. 1.

Исходное изображение	Train fonts:116 noisy	Tesseract
ЕСЛИ ЖИЗНЬ ТЕБЯ ОБМАНЕТ...	если жизнь тебя обманет	ЕСЛИ ЖИЗНЬ ТЕБЯ ОБМАНЕТ
Со мною вот что происходит Ко мне мои старые друг не ходит А ходят в праздной суете Разнообразные не те И он не с теми ходит где то И тоже понимает это И наш раздор необъясним Мы оба мучаемся с ним	Со мною вот что происходит Ко мне мои бары друг не ходит А ходят в праздной суете Разнообразные не те И он не с теми ходит где то И тоже понимает это И наш раздор необъясним Мы оба меемся с ним	Со мною вот что происходит Ко мне мои старые друг не ходит А ходят в праздной суете Разнообразные не те И он не с теми ходит где то И тоже понимает это И наш раздор необъясним Мы оба мучаемся с ним

Таблица 1. Результаты распознавания текста двумя методами

Test Train	duplicate orig	duplicate noisy	other orig	other noisy
fonts:10 orig	0.936	0.88	0.9	0.82
fonts:20 orig	0.96	0.913	0.925	0.855
fonts:30 orig	0.953	0.9	0.93	0.859
fonts:10 noisy	0.93	0.9	0.895	0.84
fonts:20 noisy	0.966	0.92	0.92	0.86
fonts:30 noisy	0.954	0.92	0.94	0.87

Таблица 2. Точность классификации. Размер шрифта на обучении — 50pt, на тесте — 30pt.

Test Train	duplicate orig	duplicate noisy	other orig	other noisy
fonts 10 orig	p(э з)=0.2 p(л п)=0.15 p(і т)=0.15	p(з э)=0.28 p(о д)=0.2 p(а д)=0.18	p(а д)=0.44 p(з э)=0.3 p(ч ц)=0.27	p(а д)=0.5 p(з э)=0.49 p(ч ц)=0.33
fonts 20 orig	p(л п)=0.1 p(з э)=0.1 p(о д)=0.1	p(з э)=0.31 p(о д)=0.14 p(п л)=0.13	p(з э)=0.29 p(а д)=0.17 p(п л)=0.16	p(з э)=0.51 p(а д)=0.24 p(п л)=0.21
fonts 30 orig	p(з э)=0.2 p(л п)=0.1 p(п л)=0.1	p(з э)=0.35 p(л п)=0.14 p(а д)=0.13	p(з э)=0.29 p(а д)=0.21 p(п л)=0.14	p(з э)=0.51 p(а д)=0.29 p(п л)=0.19
fonts 10 noisy	p(і т)=0.45 p(л п)=0.25 p(з э)=0.2	p(з э)=0.28 p(л п)=0.22 p(о д)=0.2	p(і т)=0.44 p(а д)=0.36 p(з э)=0.3	p(з э)=0.49 p(а д)=0.48 p(ч ц)=0.32
fonts 20 noisy	p(з э)=0.25 p(л п)=0.125 p(о д)=0.1	p(з э)=0.39 p(л п)=0.15 p(п л)=0.125	p(з э)=0.36 p(а д)=0.19 p(п л)=0.16	p(з э)=0.57 p(а д)=0.25 p(п л)=0.22
fonts 30 noisy	p(з э)=0.26 p(п л)=0.12 p(л п)=0.1	p(з э)=0.36 p(л п)=0.14 p(о д)=0.12	p(з э)=0.34 p(а д)=0.17 p(п л)=0.14	p(з э)=0.5 p(а д)=0.2 p(п л)=0.16

Таблица 3. Топ-3 ошибок. Размер шрифта на обучении — 50pt, на тесте — 30pt.

	L	L'
Train fonts:116 noisy	7.625	0.0753
Tesseract	2.125	0.015

Таблица 4. Средние расстояния Левенштейна и относительного Левенштейна

Анализ результатов

Из экспериментов 1 и 2 видно:

1. Больше шрифтов в обучающей выборке $\Rightarrow$ качество распознавания выше.
2. Добавление зашумленных объектов в выборку увеличивает качество классификации.
3. Для успешного распознавания букв в каких-то шрифтах необязательно, чтобы в обучающей выборке были примеры букв из этого шрифта. Это подтверждает свойство графемы быть инвариантной к стилю начертания букв.
4. Чаще всего путаются буквы, схожие по скелетной структуре, например з и э, і и т, п и м, А и Д и др.
5. Чем больше в выборке шрифтов, тем процент ошибок классификации становится меньше.

В эксперименте 3 предложенный метод немного проигрывает tesseract. Но результаты являются приемлемыми: среднее относительное редакторское расстояние порядка 0.07. На редакторское расстояние сильно влияет плохая сегментация букв: случается так, что две буквы, стоящие рядом, могут при бинаризации образовывать одну компоненту связности. Тогда два символа будут считаться за один, и распознавание предложенным методом может иметь ошибочные результаты.

8. Выводы

Эксперименты показали, что предложенная математическая модель является работоспособной: из нее можно выделить признаки, необходимые для качественного распознавания букв в изображениях.

Достоинства предлагаемого подхода:

1. Независимость от размера и шрифта буквы.
2. Выделение общей структуры (математической модели графемы) букв, которой достаточно для распознавания букв в новых шрифтах.
3. Интерпретируемые признаки и отсутствие отказов от классификации.
4. Интуитивно понятные построение признакового пространства и алгоритм классификации.
5. Наличие методов предобработки изображения с целью улучшения качества бинаризации.
6. Необязательно обучаться на большой выборке.

Недостатки предлагаемого подхода:

1. Зависимость качества распознавания от качества бинаризации и сегментации изображения.
2. Наличие ошибок классификации схожих по графемному представлению букв.

9. Заключение

В данной работе предложена улучшенная формализация понятия «графема» [11], а именно: новый способ построения математической модели графемы, удовлетворяющей свойствам графемы. На основе этой модели построен легко интерпретируемый метод генерации признаков, использующихся для применения алгоритма классификации букв в изображениях. То есть, определяется мера сходства между математическими моделями графем. Также предложен подход к предобработке и бинаризации цветных изображений и извлечения из него символов, строк и слов. Алгоритм показывает сравнимые с современными методами результаты, что подтверждают проведенные эксперименты.

10. Благодарности

Работа выполнена при поддержке РФФИ, грант 17-01-00917.

11. Литература

- [1] *Felix Hausdorff*. Grundzüge der mengenlehre. — 1914.
- [2] Hochreiter, S. Lstm can solve hard long time lag problems / Sepp Hochreiter, Jürgen Schmidhuber //

Advances in neural information processing systems. — 1997. — P. 473–479.

- [3] Ito, K. Gaussian filter for nonlinear filtering problems / Kazufumi Ito // Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187) / IEEE. — Vol. 2. — 2000. — P. 1218–1223.
- [4] Kam, H. T. Random decision forests / Ho Tin Kam // Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC. — 1995.
- [5] Kimura, F. Handwritten numerical recognition based on multiple algorithms / Fumitaka Kimura, Malayappan Shridhar // Pattern recognition. — 1991. — Vol. 24, no. 10. — P. 969–983.
- [6] Orieux, F. Bayesian estimation of regularization and point spread function parameters for wiener-hunt deconvolution / François Orieux, Jean-François Giovannelli, Thomas Rodet // JOSA A. — 2010. — Vol. 27, no. 7. — P. 1593–1607.
- [7] Otsu, N. A threshold selection method from gray-level histograms / N. Otsu // IEEE Trans. Sys., Man., Cyber. — 1979.
- [8] *ParaType*. Цифровые шрифты. — ParaType. — 2008.
- [9] Takahashi, H. A neural net ocr using geometrical and zonal pattern features / Hiroyasu Takahashi // Proc. 1st Intl. Conf. on Document Analysis and Recognition. — 1991. — P. 821–828.
- [10] Tesseract. — <https://github.com/tesseract-ocr/tesseract>.
- [11] *A. Lipkina, L. Mestetskiy*. Grapheme approach to recognizing letters based on medial representation / *A. Lipkina, L. Mestetskiy*. — 2019. — 01. — P. 351–358.
- [12] *О. В. Осетрова*. СЕМИОТИКА ШРИФТА. — Вестник Воронежского государственного университета. Серия: Филология. Журналистика. — 2006.
- [13] Yujian, L. A normalized levenshtein distance metric / Li Yujian, Liu Bo // IEEE transactions on pattern analysis and machine intelligence. — 2007. — Vol. 29, no. 6. — P. 1091–1095.
- [14] *Л. М. Местецкий*. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. — М.: ФИЗМАТЛИТ. — 2009.

Об авторах

Липкина Анна Львовна, магистр 1 года обучения кафедры математических методов прогнозирования факультета вычислительной математики и кибернетики Московского государственного университета. lipkina96@mail.ru.

Местецкий Леонид Моисеевич, д.т.н., профессор кафедры математических методов прогнозирования факультета вычислительной математики и кибернетики Московского государственного университета. mestlm@mail.ru.