

## Метод распознавания шрифтов на основе медиального представления\*

А.Л. Липкина<sup>1</sup>[0000-0002-7177-6450], Л.М. Местецкий<sup>2</sup>

<sup>1</sup> Московский государственный университет имени М.В. Ломоносова, Москва, Россия  
lipkina96@mail.ru, mestlm@mail.ru

**Аннотация.** В статье описывается метод распознавания шрифтов на основе медиального представления, интегрированный в систему распознавания шрифтов по цифровому изображению текста. Эта система ищет похожие шрифты, упорядоченные по схожести, на шрифт, изображенный на введенном пользователем изображении текста. Работа системы основана на решении двух задач машинного обучения: распознавания текста на изображении и распознавания шрифта по изображению текста. Для решения первой задачи используется понятие математической модели графемы, основанной на непрерывном медиальном представлении символа. Решение задачи распознавания шрифта основано на понятии морфологической ширины фигуры, также тесно связанной с медиальным представлением. Мы предлагаем метод использования функции морфологической ширины для поиска наиболее похожих шрифтов из известной базы. Проведенные эксперименты показывают высокую точность поиска наиболее похожих шрифтов. Для базы, состоящей из 2543 шрифтов, точность составляет 0.991 по метрике  $\text{top}@5$  для правильно распознанного текста в размере шрифта 100 пикселей на изображении.

**Ключевые слова:** графема, морфологическая ширина, цифровое изображение текста, цифровой шрифт, медиальное представление.

### 1 Введение

В мире существует огромное количество различных шрифтов. У графических дизайнеров, художников, предпринимателей и других людей часто возникает необходимость определить шрифт, в котором слово изображено на картинке, сканированном изображении или фотографии или найти похожие на него шрифты. Причины могут быть разные, например, проверить, что на шрифт есть лицензия или найти похожий свободный шрифт. Сейчас существуют форумы, на которых люди могут вывесить свою картинку со шрифтом и попросить кого-нибудь определить шрифт, но не всегда можно получить таким образом ответ. Как другой вариант, можно попросить эксперта определить шрифт на изображении, однако ввиду большого визуальный метод определения шрифтов может

---

\* Работа выполнена при поддержке гранта РФФИ № 20-01-00664.

приводить к ошибкам. Поэтому для данной задачи важны автоматизированные системы распознавания шрифтов.

В настоящее время существует ряд web-сайтов, например: WhatTheFont\*, WhatFontIs†, FontSquirrel‡, Identifont§, которые способны распознавать шрифты по загруженному изображению. Однако не все из них являются удобными с точки зрения пользователя (сложный диалог взаимодействия с пользователем) и не все приспособлены к определению кириллических шрифтов. В дополнение к этому сами алгоритмы в сервисах закрытые, что создает неудобства для исследователя в области распознавания шрифтов.

Статей по рассматриваемой тематике довольно мало, и тем более еще меньше статей, в которых рассматриваются методы, умеющие распознавать шрифт по большой базе различных шрифтов. В большинстве из самых цитируемых статей [9, 8] тестирование методов проводится на маленьких выборках (20-50 шрифтов) и результаты экспериментов показывают высокую точность распознавания. В некоторых статьях [1, 2, 7] рассматриваются большие базы тестируемых шрифтов (1000 - 2000 шрифтов), однако в них по результатам экспериментов наблюдается меньшая точность по сравнению с предыдущими перечисленными статьями.

В данной статье мы предлагаем метод распознавания шрифтов (см. рис. 1), заложенный в системе распознавания шрифтов, которая умеет распознавать текст на изображении и находить похожие шрифты из базы, состоящей из 2543 шрифтов, поддерживающих кириллический алфавит [13], причем от и до она завязана на морфологических признаках, характеризующих структуру символа. Система умеет не только определять изображенный на картинке шрифт, но и выдавать похожие на него шрифты из выбранной базы в виде списка изображений шрифтов. Изображения в списке отсортированы по убыванию степени похожести шрифтов на действительный шрифт.

## 2 Постановка задачи

На вход программе подается изображение с текстом. Необходимо определить наиболее похожие шрифты из выбранной базы эталонных шрифтов и упорядочить их по степени убывания похожести. Формально задача состоит из следующих шагов:

- Обработка изображения.
- Сегментация изображения с целью выделения глифов.
- Распознавание символа в каждом сегментированном фрагменте.
- Поиск похожих шрифтов для каждой классифицированной буквы.

---

\* <https://www.myfonts.com/WhatTheFont>

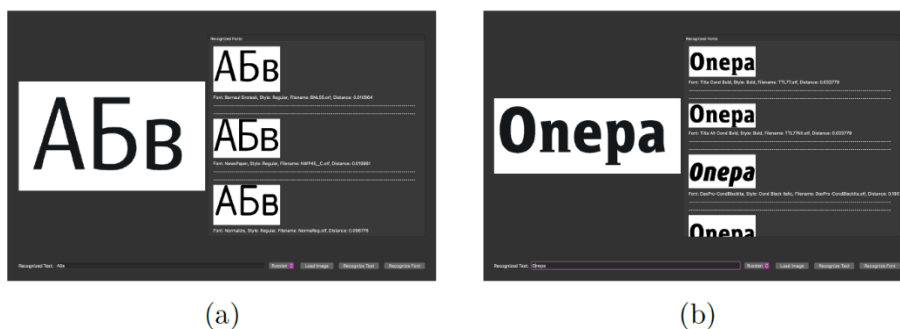
† <https://www.whatfontis.com/all-fonts.html>

‡ <https://www.fontsquirrel.com/matcherator>

§ <http://www.identifont.com/>

- Упорядочивание шрифтов по степени похожести на изображенный шрифт. Эта сортировка основана на информации о найденных шрифтах для каждой отдельной буквы.
- Создание графического интерфейса, позволяющего пользователю загрузить изображение и получить ранжированный список наиболее похожих шрифтов.

Методы обработки и сегментации осуществляются в соответствии с выбранным форматом входных данных, поэтому в данной статье они не описываются.



**Рис. 6.** Демонстрация разработанной системы распознавания шрифтов. В обоих случаях правильно распознаны символы и найдены похожие шрифты (в том числе и шрифт на изображении на 1-м месте выдачи)

### 3 Метод решения

Решение задачи распознавания шрифта по цифровому изображению основано на двух элементах: распознавание символов на изображении и распознавание шрифта на изображении известной буквы. Подход к распознаванию символов описан в [4, 12], а построение признаков, используемых в решении задачи распознавания шрифта, описано в [11].

Алгоритм распознавания символов основан на построении непрерывного скелетного графа для изображения буквы [10], выделении признаков по расположению вершин [12] в полученном графе и последующем обучении классификатора. Построение признаков, используемых в задаче распознавания шрифта, базируется на анализе гистограммы морфологической ширины [5, 11], которая показывает зависимость площади дискового покрытия фигуры от радиуса покрывающих дисков.

#### 3.1 Метод распознавания символов

В основе решения задачи распознавания символов на цифровом изображении лежит понятие математической модели графемы [4, 12] и метод выделения при-

знаков из нее для последующей возможности классификации символа при помощи алгоритмов машинного обучения.

**Графема** — это графический примитив, геометрический граф, универсальный образ буквы, понятный любому грамотному человеку и используемый в обучении грамоте. Он является основой любого шрифта и может быть представлен в виде контура букв в тонком шрифте (см. рис. 2). Эти свойства графемы используются для построения универсального алгоритма распознавания символов.

На содержательном уровне графемы обладают следующими свойствами:

5. Любые две графемы хорошо отличимы друг от друга.
6. Если два изображения представляются одной графемой, то различия между изображениями несущественны (по структуре графемы).

Основной идеей для построения математической модели графемы является построение *непрерывного представления* скелетного графа бинарного изображения буквы и дальнейшее выделение существенного подграфа из него, отражающего свойства букв. Процесс построения можно увидеть на рис. 3.



Рис. 7. Примеры букв в различных шрифтах (слева) и их графем (справа)



Рис. 8. Процесс построения графемы, слева направо: бинарное изображение символа; аппроксимирующий многоугольник; вписанные пустые круги фигуры; скелетный граф; граф с удаленными по некоторому принципу рёбрами; метаграф - математическая модель графемы

В конечном счете математическая модель графемы представляет собой метаграф  $S$ , вершины которого могут иметь только степени 1 и 3, а метаребра которого - «цепочки» последовательных ребер, в которых концевые вершины цепочек имеют степень 1 или 3, а остальные - степень 2. Далее из метаграфа выделяются признаки, которые впоследствии используются в классификаторе символов.

### 3.2 Метод распознавания шрифтов

В основе метода лежит понятие функции морфологической ширины фигуры. Быстрый и эффективный способ вычисления данной функции был предложен в [11]. Приведем необходимые определения и кратко опишем используемое понятие.

**Дисковое  $r$ -покрытие фигуры** - объединение всех пустых кругов радиуса  $r$ .  
 **$r$ -площадь фигуры** - площадь её дискового  $r$ -покрытия.

**Морфологическая ширина фигуры** - это  $r$ -площадь фигуры как функция от  $r$  (см. рис. 4).

Заметим, что морфологическая ширина является невозрастающей функцией от  $r$ . Примеры различных функций морфологической ширины и соответствующие им изображения букв можно увидеть на рис. 5. Также отметим, что в нашей постановке эта функция является дискретной.

#### Описание предлагаемого подхода

Сначала приведем несколько обозначений. Пусть  $DB$  - зафиксированная база шрифтов,  $H_{DB}$  - высота шрифта в базе,  $R_{DB}$  - шаг дискретизации (радиуса), с которым строятся все функции морфологической ширины в базе шрифтов,  $F(L)$  - высота шрифта, в котором изображена буква  $L$ ,  $H(L)$ ,  $W(L)$  - непосредственно высота и ширина ограничивающей рамки изображенной буквы,  $SQ(L)$  - количество черных пикселей на изображении буквы (площадь буквы). Также, пусть  $SP_r(L)$  - функция морфологической ширины с шагом радиуса  $r$  для буквы  $L$ ,  $SP_r(L)|_x$  - значение морфологической ширины в точке  $x$ ,  $R_{max}(L)$  - максимальный радиус пустого вписанного круга в фигуру буквы  $L$ ,  $Dom(R, r)$  - равномерная сетка на отрезке  $[0, R]$  с шагом радиуса  $r$ ,  $\#Dom(R, r)$  - число точек в сетке, аналогично  $Dom(SP_r(L))$  - носитель морфологической ширины,  $\#SP_r(L)$  - число точек носителя функции морфологической ширины.

В общем виде схема выглядит следующим образом:

- Выбираем базу шрифтов; для каждого шрифта и каждой буквы из фиксированного алфавита строим морфологическую ширину, полагая размер шрифта  $H_{DB}$  и шаг радиуса  $R_{DB}$  для всех случаев одинаковый (размер шрифта = размер заглавной буквы «Н»).
- Для тестового изображения символа вычисляем морфологическую ширину с некоторым шагом.
- Из всей базы эталонных шрифтов выделяем подмножество кандидатов  $E_F$  на звание похожих шрифтов.
- Сравниваем морфологическую ширину символа со всеми функциями морфологической ширины из  $E_F$  для этой буквы.
- Сравнение этих функций производится посредством метрики  $L_I$ .
- Все шрифты в базе упорядочиваются на основе значений линейной комбинации некоторых функций от метрик всех букв.



Рис. 9. Демонстрация дисковых  $r$ -покрытий фигур для разных значений  $r$

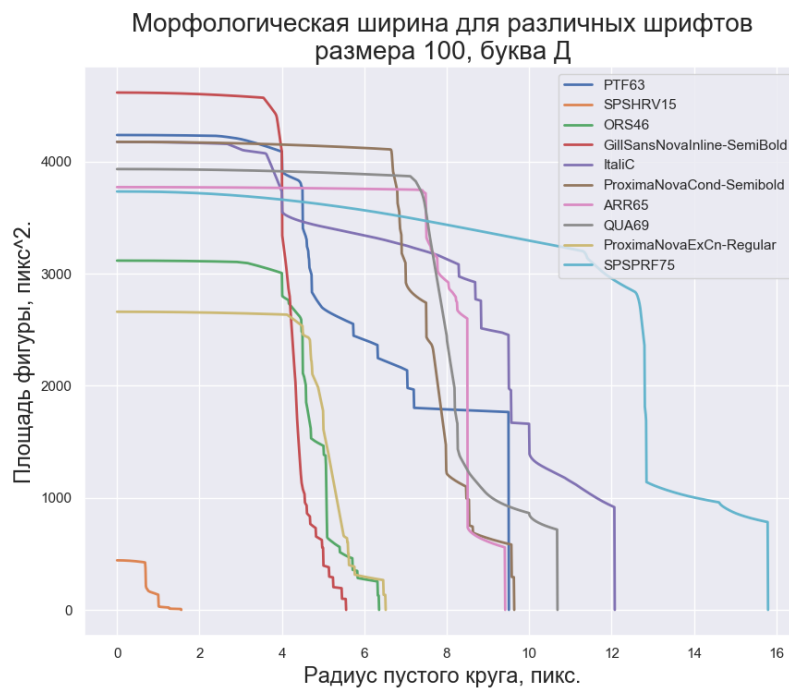


Рис. 10. Примеры морфологических ширин на равномерной сетке в различных шрифтах

### Выделение кандидатов близких шрифтов

Для больших баз шрифтов сравнение функций морфологической ширины тестового слова со всеми шрифтами из базы может быть дорогостоящей операцией: подсчет метрик имеет сложность  $O(NW)$ , где  $N$  - число шрифтов в базе,  $W$  - число распознанных букв в слове. Также для упорядочивания используется сортировка, работающая за  $O(N \log N)$ . Поэтому мы хотим среди всей базы оставить некоторое подмножество шрифтов-кандидатов, которые являются похожими на шрифт на картинке по некоторым статистическим и быстро вычисляемым признакам.

Для шрифтов нет универсальных стандартов пропорций изображенных в них букв. Количество черных пикселей (площадь буквы) при одной и той же высоте шрифта для разных типов шрифтов также разнится. В качестве легко вычисляе-

мых статистических признаков шрифта мы выделяем 4 признака для изображения буквы  $L$  (1).

$$\left[ \frac{H(L)}{W(L)}, \frac{SQ(L)}{H(L) \cdot W(L)}, \frac{SQ(L)}{H(L)^2}, \frac{SQ(L)}{W(L)^2} \right]. \quad (5)$$

На этапе обучения для каждого символа алфавита из базы строятся вышеприведенные признаки, нормируются при помощи z-нормализации [6] и на полученных данных обучаются модели  $K$  ближайших соседей (k-NN) [3].

На этапе тестирования для изображения вычисляются и нормируются признаки и при помощи соответствующего букве классификатора k-NN ищутся  $K_F$  ближайших соседей, где  $K_F$  – заранее выбранное число. Далее, для каждого шрифта из полной базы определяется, оказался ли этот шрифт среди  $K_F$  хотя бы для  $p\%$  от числа букв в слове. Если это так, то шрифт добавляется в множество шрифтов-кандидатов. В конце концов алгоритм сравнения функций морфологической ширины будет работать только с множеством шрифтов-кандидатов.

### Сравнение функций морфологической ширины

При масштабировании одного и того же изображения в  $k$  раз, морфологические ширины изменяются в  $k$  раз по значениям их носителя и в  $k^2$  раз по значениям площадей покрытия. Для корректного сравнения данных функций необходимо, чтобы число точек, на которых задана эта функция, было одинаковым, а также чтобы совпадали размеры шрифтов на тестовом символе и в базе эталонных шрифтов. Пусть  $L$  - тестовая буква в некоторой высоте шрифта  $F(L)$ . Для удачного масштабирования морфологической ширины тестового изображения  $L$  необходимо знать размер шрифта  $F(L)$  тестового символа. Если мы знаем размер шрифта, то сможем рассчитать коэффициент масштабирования на основе размеров этих шрифтов  $s_{DB,L} = \frac{H_{DB}}{F(L)}$ , а также рассчитать сам шаг радиуса  $r'$  для построения  $SP_{r'}(L)$ :  $r' = \frac{R_{DB}}{s_{DB,L}}$ . Далее возможно корректное сравнение функций морфологической ширины посредством умножения на коэффициент масштабирования, описанный в начале секции.

Но на практике не всегда возможно узнать размер шрифта в тестовом изображении: например, если всё изображение состоит из одного символа, скажем, «с», то нельзя с уверенностью сказать, является ли это заглавной или строчной буквой. Поэтому на практике можно столкнуться с тем, что у морфологической ширины в базе эталонов и морфологической ширины тестового изображения будет разное число точек в носителе.

Из-за этого возникает отдельная задача построения морфологической ширины для тестового изображения буквы и ее корректного сравнения с аналогичными функциями из базы эталонных шрифтов. Для решения этой проблемы мы предлагаем следующий подход. В базе эталонов ищется максимальная морфологическая ширина по числу точек определения (2).

$$N_{max} := \max_{L \in DB} \#SP_{R_{DB}}(L). \quad (6)$$

Затем для тестового изображения символа  $L$  строим морфологическую ширину  $SP_r(L)$ , определенную в  $N_{max}$  равномерных точек на отрезке  $[0, R_{max}(L)]$  ( $r' = \frac{R_{max}(L)}{N_{max}-1}$ ). Далее сравниваем  $SP_r(L)$  с функцией морфологической ширины  $SP_{r_{DB}}(M)$  символа некоторого  $M$  из  $DB$ , определенной в  $K := \#SP_{r_{DB}}(M)$  равномерных точках. Для корректного сравнения мы будем линейно интерполировать  $SP_r(L)$  на  $K$  равномерных точек. Так как линейная интерполяция сохраняет монотонность, то полученное приближение также будет некоторой функцией морфологической ширины. После приведения к одному носителю, можно рассчитать коэффициент масштабирования площадей для  $SP_r(L)$  как  $s_{M,L} := \frac{H(M)}{H(L)}$ . Так как теперь количество точек в носителях совпадает, то перенумеруем их на  $Dom(K - 1, 1)$ . Расстояние между функциями считается по метрике  $L_1$  (3).

$$d(M, L) = \sum_{x \in Dom(K-1, 1)} (|SP_1(L)|_x - |SP_1(M)|_x). \quad (7)$$

Итоговое расстояние между шрифтом слова  $W$  и шрифтом из базы  $F$  считается по всем буквам  $L_i$  слова  $W$  по следующей формуле (4).

$$dist(W, F) = \frac{1}{|W|} \sum_{i=1}^{|W|} (1 - e^{-d(M_i, L_i)}), \quad (8)$$

где  $|W|$  – число букв в слове  $W$ , а  $M_i$  изображает ту же букву, что и  $L_i$ , но в шрифте  $F$ .

### Метрика качества

На этапе тестирования мы используем метрику качества  $Stop@k$  для различных значений  $k$ . Пусть есть упорядоченный по некоторому принципу список классов  $C = \{c_1, c_2, c_3, \dots, c_{|C|}\}$ , его длина  $|C|$ . Тогда метрика  $top@k$  для правильного ответа  $c'$  считается вычисляется по формуле (5).

$$top@k(c', C) = I[c' \in \{c_1, \dots, c_k\}]. \quad (9)$$

На тестовой выборке, состоящей из нескольких слов, метрика качества считается как среднее всех  $top@k$  для входящих в нее слов.

## 4 Вычислительные эксперименты

### 4.1 Используемое программное обеспечение

Построение скелетного графа и функции морфологической ширины бинаризованного изображения реализовано на языке C++. Построение математической модели графемы, предлагаемый в работе алгоритм классификации и метод поиска наиболее схожих шрифтов были реализованы с помощью средств языка



Python3. Графический интерфейс системы реализован с помощью библиотеки QT\* на Python3.

#### 4.2 Эксперименты на качество системы распознавания шрифтов

В экспериментах мы тестируем качество распознавания шрифтов, используя предлагаемый метод. Эксперименты проводились на базе из 2543 шрифтов компании Paratype [13], среди которых система будет определять наиболее похожие шрифты, и упорядочивать их по степени схожести.

Модель распознавания символов обучается на 34 буквах кириллического алфавита («ы» считается за две графемы: «i» и «ь») в 2 начертаниях (строчное и прописное) из базы шрифтов. Размеры символов в тренировочном множестве берутся равными 120 и 70, изображения генерируются как в зашумленной (при помощи фильтра Гаусса с дисперсией 1), так и в не зашумленной форме, а количество классов в задаче классификации полагается равным 34.

Значения параметров полагаем следующими:

- $H_{DB} = 1000$  пикс;
- $R_{DB} = 0.1$  пикс;
- $K_F = 70$ ;
- $p = 50$ .

#### Сравнение качества для разных стратегий выбора шрифтов-кандидатов

В данном эксперименте мы сравниваем качество распознавания шрифтов при разных множествах шрифтов-кандидатов. Для создания тестовой выборки проводилась генерация примерно 500 случайных слов произведения Л.Н.Толстого «Война и мир», изображенных в 7 случайных шрифтах из доступной базы  $DB$  в размерах шрифта 100 пикселей.

Для случая, обозначаемого через «Native», наиболее похожие функции морфологической ширины искались по всей базе эталонных шрифтов  $DB$ . В случае, обозначаемом через «kNN» мы использовали поиск множества шрифтов-кандидатов при помощи алгоритма поиска ближайших соседей; в этом случае наиболее похожие шрифты искались только среди отобранных шрифтов. Пометка «All» означает, что качество считалось по всем изображениям в тестовой выборке. Пометка «Good» означает, что качество считалось по тем словам, истинная длина которых и предсказанная длина совпадали. Пометка «Right» означает, что качество считалось только по тем словам, на которых алгоритм распознавания не ошибся. Результаты можно видеть на рис. 6.

---

\* <https://www.qt.io/>

### Исследование попадания шрифтов в множество шрифтов-кандидатов на стадии использования kNN

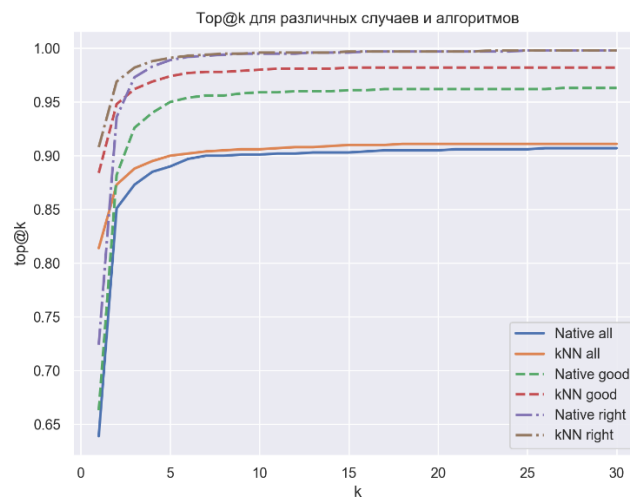
В данном эксперименте исследуется процент попадания правильного шрифта во множество шрифтов-кандидатов, которых не более  $K_f=70$  шрифтов. Исследование проводилось на той же выборке, что и в предыдущем эксперименте. Результаты можно увидеть в таблице 1.

**Таблица 3.** Точность попадания правильного шрифта в отобранное множество шрифтов-кандидатов

	kNN All	kNN Good	kNN Right
Точность	0.907	0.987	0.999

### Исследование зависимости качества от размера шрифта и качества тестовой выборки

В данном эксперименте проводится исследование зависимости качества от размера шрифта в тестовой выборке и качества тестовой выборки. Генерация тестовой выборки производится так же, как и ранее. Всего создается 3 тестовых выборки. Одна из них в размере шрифта 100 и не зашумлённая («100 pure»), другая - в размере шрифта 150 и не зашумленная («150 pure») и третья в размере шрифта 150 и зашумленная («150 spoil»). Результаты находятся на рис. 7.



**Рис. 11.** Визуализация метрик  $top@k$  для  $k=1..30$  для различных стратегий выбора наиболее похожих шрифтов

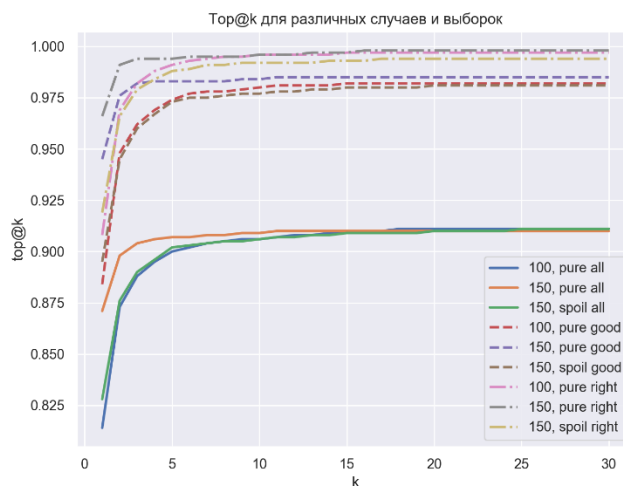


Рис. 12. Визуализация метрик  $top@k$  для  $k=1..30$  для различных высот шрифтов в выборках и различных качеств изображений

## 5 Выводы

Эксперименты подтвердили работоспособность предлагаемого в данной работе метода: для  $top@5$  уже достигаются качества 0.9, 0.974, 0.991 для всех шрифтов, для шрифтов, в которых распознанное слово по длине равно слову-ответу, для шрифтов, в которых алгоритм распознавания текста выдал точный ответ соответственно. Также из экспериментов видно, что метод выделения шрифтов-кандидатов значительно улучшает качество. Хотя и текущее значение  $K_F=70$  не является оптимальным, подбор наилучшего с точки зрения метрик значения  $K_F$  открывает возможности для будущих исследований в области улучшений работы метода. Алгоритм является достаточно устойчивым к размеру шрифта на изображении и качеству изображения: по метрике  $top@10$  метод показывает примерно одинаковое качество распознавания в перечисленных случаях, однако по  $top@1$  качество распознавания в большем размере шрифта (150) существенно выше, чем в меньшем (100).

## 6 Заключение

В данной статье был предложен метод распознавания шрифтов по цифровым изображениям, показывающей высокие результаты распознавания для отдельных классов шрифтов. Метод основан на вычислении функции морфологической ширины в контексте задачи поиска наиболее похожих шрифтов и способе сравнения данных функций. Также была разработана система распознавания

шрифтов с графическим интерфейсом. Эксперименты показали высокие результаты качества распознавания шрифтов на изображении.

Работа выполнена при финансовой поддержке гранта РФФИ, проект 20-01-00664 «Морфологический анализ изображений и видеопоследовательностей на основе непрерывного медиального представления и машинного обучения».

## Литература

1. Bui, T., Collomosse, J.: Font finder: Visual recognition of typeface in printed documents. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 3926–3930. IEEE (2015) Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
2. Chen, G., Yang, J., Jin, H., Brandt, J., Shechtman, E., Agarwala, A., Han, T.X.: Large-scale visual font recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3598–3605 (2014).
3. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE transactions on information theory 13(1), 21–27 (1967).
4. Lipkina, A., Mestetskiy, L.: Grapheme approach to recognizing letters based on medial representation. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019). vol. 4, pp. 351–358. Setubal, Portugal (2019).
5. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Trans. Pattern Anal. Mach. Intell. 11, 701–716 (1989).
6. Maxfield, M., Schweitzer, J., Gouvier, W.: Measures of central tendency, variability, and relative standing in nonnormal distributions: alternatives to the mean and standard score. Archives of physical medicine and rehabilitation 69(6), 406–409 (1988).
7. Wang, Z., Yang, J., Jin, H., Shechtman, E., Agarwala, A., Brandt, J., Huang, T.S.: Deepfont: Identify your font from an image. In: Proceedings of the 23<sup>rd</sup> ACM international conference on Multimedia. pp. 451–459 (2015).
8. Zhu, Y., Tan, T., Wang, Y.: Font recognition based on global texture analysis. IEEE Transactions on pattern analysis and machine intelligence 23(10), 1192–1200 (2001).
9. Zramdini, A., Ingold, R.: Optical font recognition using typographical features. IEEE Transactions on pattern analysis and machine intelligence 20(8), 877–882 (1998).
10. Местецкий Л.М.. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. М.: ФИЗМАТЛИТ (2009).
11. Липкина А., Ломов Н., Местецкий Л.. Распознавание шрифтов по изображениям на основе анализа структуры букв // ГрафиКон-2017 Труды 27-й Международной научной конференции. Пермский государственный национальный исследовательский университет. С. 359-366. (2017).
12. Липкина А., Местецкий Л.. Структурный подход к классификации букв в изображениях // ГрафиКон-2019 Труды 29-й Международной конференции по компьютерной графике и машинному зрению. Брянский государственный технический университет. С. 136-141. (2019).
13. Якупов Э., Петрова И., Фридман Г., Королькова А., Левин Б.. 2008-2014. ПАРАТАЙП. Цифровые шрифты. Каталог. Москва: ParaType (2014).