Анализ машинописных подписей к фотографиям в цифровом альбоме

А.Н. Талбонен, А.А. Рогов

Петрозаводский государственный университет

perhetal@onego.ru, rogov@psu.karelia.ru

Аннотация

Статья посвящена вопросам формирования электронной коллекции фотографий строительства Беломорско-Балтийского канала. В статье описаны первые этапы проделанной работы, а именно, считывание подписей к фотографиям и оценка качества их распознавания.

1 Особенности задачи

Рассматриваемая в данной статье задача возникла вследствие необходимости организации поиска в большом массиве цифровых фотографий, относящихся к одной более крупной тематике. В качестве исходного материала выступила коллекция снимков строительства Беломорско-Балтийского канала (ББК), сделанных в 1930-е годы. Данная коллекция состоит из 8-ми альбомов в среднем по 800 снимков в каждом, что в общей сложности составляет почти 6,5 тыс. изображений, и находится в Карельском государственном краеведческом музее. Каждое изображение данной коллекции представляет собой сфотографированный лист, на который были наклеены оригинальная фотография, а также подпись к фотографии в виде небольшой бумажной полоски, с машинописным текстом. Данный текст содержит информацию о времени, месте нахождения объекта снимка, кроме того, кратко описываются содержание объекта и сюжет. Помимо фотографии подпись также несет в себе определенную ценность. В частности, приведение конкретной подписи в электронный текст с последующим логическим разбиением на составляющие: номер, время, место, сюжет, объект, – позволит выполнить классификацию изображений данной коллекции по любому из данных признаков. Таким образом, пользователь сможет фильтровать коллекцию и находить только интересующие его изображения, указывая в поиске нужные параметры.

На процесс преобразования подписей к фотографиям в электронно-текстовую форму для последующего анализа существенно повлияли следующие факты:

- оригинальные снимки и подписи к ним были сделаны достаточно давно, что негативно сказалось на их качестве;
- цифровые изображения были получены методом фотографирования, при этом было изначально использовано достаточно низкое разрешение: 75 dpi;
- полученные снимки были сжаты в формат JPEG, что также сказалось на качестве изображений;
- все изображения – черно-белые.

Типичный пример изображения можно увидеть на рис. 1.

Рис. 1. Пример изображения коллекции, посвященной строительству ББК

В данных условиях возникает целый ряд проблем, связанных с распознаванием текста подписей.

1. Подписи на цифровых изображениях характеризуются низким качеством, а именно:
   a. на текстовом фоне присутствует шум, который обусловлен как изношенностью бумаги, так и неизбежно возникающей размытостью при сжатии растрового изображения в JPEG;
   b. аналогичный шум наблюдается на участках литер;
   c. отсутствует резкость линий букв текста;
   d. ровни серого текста и фона на некоторых фотографиях отличаются незначительно;
2. Распознавать текст на изображениях, подставляя их в готовую систему распознавания, например, FineReader, крайне затруднительно из-за того, что OCR в некоторых случаях ошибочно рас-

Труды 12[й] Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2010, Казань, Россия, 2010

познает элементы участка на самой фотографии как отдельные символы, что приводит к многочисленным ошибкам и возникновению мусора. Кроме того, встречаются случаи, когда OCR вообще не находит области с текстом на фоне фотографии либо из всего текста обнаруживается и распознается только одна часть вследствие неправильного определения границ области текста.

Таким образом, все вышеперечисленные проблемы вынуждают искать различные пути улучшения качества как результатов, так и исходных данных.

2 Анализ непосредственного решения задачи

Для проведения непосредственного анализа OCR были отобраны 12 изображений с характерными недостатками, которые были описаны в предыдущем разделе. Ниже представлены основные результаты непосредственного распознавания текста.

Таблица 1. Результаты прямого распознавания изображений

№	W_j^n	N_j^o	I_j^r	G_j	ν_j
1	9	16	1	0	0.56
2	9	15	1	1	0.56
3	0	7	0	0	0.00
4	0	11	0	0	0.00
5	6	7	1	0	0.86
6	1	8	0	1	0.11
7	5	7	1	4	0.45
8	0	10	0	0	0.00
9	0	16	0	0	0.00
10	10	11	1	3	0.71
11	8	11	0	6	0.47
12	9	12	1	6	0.50
Итого	57	87	6	21	0.38

В табл. 1 использованы следующие обозначения:

W_j^n – общее количество правильно распознанных слов файла *j*;

N_j^o – общее количество слов на изображении;

I_j^r – индекс распознавания области текста; он равен 1 в случае, когда OCR правильно находит и выделяет область текста; в случае, когда OCR выделяет мусор или текст не полностью, пропуская определенные области, индекс равняется 0;

G_j – количество слов-мусора; к ним относятся слова, которые не могут быть результатом ошибочного распознавания оригинальных слов, т. е. те, которым на изображении не соответствует ни одно слово; данные слова возникают из-за ошибочного распознавания элементов изображения, не содержащего текст;

ν_j – доля правильно распознанных слов:

$$\nu_j = W_j^n / N_j^o .$$

Как видно из таблицы, количество файлов, которые были хоть как-то распознаны, равно 8, что составляет всего 66% от общего количества файлов. А общая доля правильно распознанных слов, которая вычисляется по формуле $\nu = \sum_j W_j^n / \sum_j N_j^o$, , составляет всего лишь 38%. Более того, только в половине случаев OCR смогла правильно выделить область текста. Основная причина таких низких результатов – наличие в процессе распознавания большой помехи в виде фотографии. Поэтому для повышения результатов распознавания целесообразно отделить область текста от остальной части изображения и распознавать только текстосодержащие области.

3 Выделение области подписи

В качестве первого улучшения качества распознавания был реализован несложный эвристический алгоритм выделения границ подписи на изображении. Основная эвристика заключается в принятии того, что в большинстве случаев область текста вплотную прилегает к одной стороне изображения, занимая ее целиком по длине, тем самым разделяя изображение на две разные по содержанию части. Кроме того, в большинстве случаев область текста представляет собой однотонный прямоугольник с расположенным на нем текстом, цвет которого резко контрастирует с цветом фона, а сам текст отстоит от нижнего края области текста (в случае, если сориентировать ее прямо) минимум на 10 пикселей, что является достаточным для определения среднего цвета фона. На основе анализа изображений было выявлено, что высота области, содержащей одну строку текста, не превышает 30 пикселей.

Рассмотрим случай, когда подпись располагается в нижней части изображения прямо под фотографией. Поскольку текст и фотография в большинстве случаев располагаются горизонтально, а цвет фона сильно контрастирует как с текстом, так и с фотографией, то для выделения границ подписи достаточно оперировать средней яркостью горизонтальных линий. Данный параметр равен среднему арифметическому значений яркостей пикселей одной конкретной горизонтальной линии. Для выявления граничной горизонтальной линии, отделяющей подпись от фотографии, был использован следующий алгоритм.

1. Будем вести отсчет снизу вверх.
2. Рассчитаем среднюю яркость первых NBr строк изображения. Полученное значение Grad приблизительно равно средней яркости всей области подписи и будет являться эталоном для сравнения с яркостями других линий.
3. Зададим порог GradBr, который определяет максимальное отклонение яркости «светлого» пикселя от средней яркости фона Grad. Если модуль разности яркости текущего пикселя и Grad будет

меньше GradBr, то данный пиксель будем считать «светлым».

4. Найдем «специальную среднюю яркость строки», которая будет равна доле «светлых» пикселей данной строки.

5. Зададим порог яркости строки BGBr, выше которого строка будет считаться «светлой», а ниже – «темной». Для наглядности можно ввести параллельное 2-е изображение с такими же размерами, что и оригинальное, и окрашивать каждую ее строку в черный / белый цвет, если соответствующая строка оригинального изображения будет «темной» / «светлой». Таким образом, на месте строк текста мы получим полосы черного цвета определенной толщины.

6. Зададим максимально допустимую толщину темных полос, соответствующих тексту, TH.

7. Первые NBr строк автоматически считаем «светлыми».

8. Далее, для каждой строки будет определять, является ли она «темной» или «светлой».

9. Будем запоминать границы полос из идущих подряд «темных» строк.

10. При достижении толщины текущей «темной» полосы выше TH алгоритм прекращаем.

11. Наиболее высокая граница «темной» полосы с толщиной не больше TH будет являться границей подписи.

12. После этого копируем область изображения «ниже» найденной границы. «Ниже» означает – от начальной линии до границы.

На рис. 2 представлена область подписи изображения на рис. 1, выделенная данным методом.

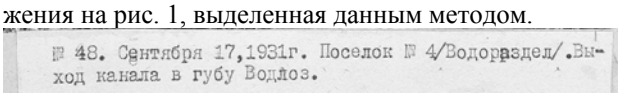
№ 48. Сентября 17,1931г. Поселок № 4/Водораздел/.Выход канала в губу Водлоз.

Рис. 2. Подпись для изображения на рис. 1

Данный алгоритм будет выполняться аналогично для случаев, когда подпись располагается в других частях. Разница будет только в направлении отсчета линий.

Ниже представлены результаты распознавания вышеуказанных 12 изображений с той разницей, что на них заранее были выделены области текста. В данном случае в OCR подаются только изображения подписей вместо целых изображений.

Таблица 2. Результаты распознавания выделенной области текста

№	W_j^n	N_j^o	I_j^r	G_j	ν_j
1	15	16	1	0	0.94
2	10	15	0	1	0.63
3	7	7	1	0	1.00
4	9	11	1	1	0.75
5	6	7	1	0	0.86
6	6	8	1	0	0.75
7	5	7	1	0	0.71
8	8	10	1	1	0.73
9	10	16	1	1	0.59
10	8	11	1	1	0.67
11	8	11	1	0	0.73
12	10	12	1	3	0.67
Итого	102	131	11	8	0.73

Как видно из табл. 2, все файлы на этот раз были распознаны, а общая доля правильно распознанных слов составила уже 73 %. При этом процент правильно выделенных областей составил 92 %. Таким образом, введение в решение данного алгоритма значительно повысило качество распознавания текста. Другим важным фактором в применении этого метода является то, что оригинальные изображения, которые используются, были предоставлены Карельским государственным краеведческим музеем и являются его собственностью. Поэтому решение обрабатывать только изображения подписей, а не сами фотографии, является приемлемым для владельца данных фотографий.

Несмотря на значительное улучшение качества распознавания, результат остается неудовлетворительным, поэтому было принято решение провести предварительную обработку изображений подписей.

4 Методы обработки изображений

Данные изображения характеризуются низким качеством. Распознаванию мешают недостаточная четкость символов текста и так называемые эффекты «соль-перец», представляющие собой хаотично расположенные пиксели с экстремальными значениями яркости (0 и ближе к 0 – «перец», 255 и ближе к 255 – «соль»). Для того чтобы избавиться от лишних шумов и повысить качество изображения, традиционно используют различные методы улучшения изображений. В данной работе были рассмотрены и опробованы 12 методов, перечисленных ниже. Для удобства описания методов улучшения изображений введем некоторые понятия:

1. Функция $f(x,y)$, для которой $x \in [0;W),\ y \in [0;H);\ x,y \in \mathrm{Z}$, где W и H – ширина и высота изображения, а $f \in \mathrm{Z}$ называется функцией изображения.

2. Функция нормирования $N(f)$:

$$N\big(f(x,y)\big) = \frac{f(x,y) - \min\limits_{i,j} f(i,j)}{\max\limits_{i,j} f(i,j) - \min\limits_{i,j} f(i,j)},$$

где $i \in [0;W),\ j \in [0;H);\ \ i,j \in \mathrm{Z}.$

3. Все арифметические операции над функциями изображений выполняются попиксельно, например:

$$f + g = h\ \forall x,y:\ h(x,y) = f(x,y) + g(x,y).$$

4. Ядро (маска) свертки M определяется как матрица размером $R \times S$ с центром в точке (r,s). Обычно $R = S$, а отсчет координат начинается с 0.

5. Функция свертки $\mathrm{Conv}(f(x,y), M, x, y))$ функции изображения $f(x,y)$ с помощью ядра

(маски) M с центром в (r,s) и фактором F определяется следующим образом [2]:

$$\mathrm{Conv}\big(f(x,y),M,x,y\big)=\frac{1}{F}\sum_{i=-r}^{R-r-1}\sum_{j=-s}^{S-s-1}f(x+i,y+j)\times \times M(i+r,j+s).$$

Будем считать, что центр ядра является центральным элементом матрицы M, а фактор $F=1,$ если иные значения не оговорены.

6. Функция изображения $h(x,y)$ называется аддитивной, если она была получена в результате арифметической операции результирующей функции $g(x,y)$ и оригинальной $f(x,y)$. Например, $h(x,y)=g(x,y)+f(x,y)$ – часто распространенный прием повышения четкости изображения за счет наложения на него результата применения Лапласиана.

7. Функция изображения $h(x,y)$ считается выровненной к оригинальной функции $f(x,y)$, если к первой было применено гистограммное выравнивание [2].

Перейдем к описанию рассмотренных в данной работе методов:

1. *Эвристический метод порогового отсечения без параметров.* Суть метода состоит в том, что на изображении устраняются все пиксели (а именно, заменяются белым цветом), величина которых равна или превышает некоторое значение. Из-за необходимости определять это значение и того, что оно может зависеть от общей яркости изображения (чем темнее фон, тем меньше должно быть пороговое значение), было решено использовать следующую формулу определения порога:

$$C=(W\,H)^{-1}\sum_{x,y}f(x,y)+D,,$$

где D – некоторое целое значение, выбираемое эмпирически и устраняющее вклад темных пикселей в оценку средней яркости фона. В нашем случае было выбрано значение 10. Таким образом, итоговая функция изображения определяется следующим образом:

$$h(x,y)=\begin{cases}f(x,y), & f(x,y)<C,\\ 0, & f(x,y)\ge C.\end{cases}$$

Обозначим данную функцию как Cut.

2. Методы пространственной фильтрации, применяющие Лапласиан [2, 3].

a. *Аддитивное изображение с использованием простого Лапласиана*. Ядро Лапласиана выглядит следующим образом:

$$M=\begin{pmatrix}-1&-1&-1\\-1&9&-1\\-1&-1&-1\end{pmatrix}.$$

Результат свертки функции $f(x,y)$ с таким ядром дает дискретный Лапласиан $g(x,y)$. Обычно диапазон значений $g(x,y)$ резко отличается по сравнению с диапазоном значений оригинальной функции, поэтому первую функцию необходимо нормировать. В результате итоговая функция будет определяться следующим образом:

$$h(x,y)=N(f(x,y)+N\mathrm{Conv}(f(x,y),M,x,y))).$$

Данная функция будет обозначена как ALaplas.

b. *Выровненное аддитивное изображение с использованием простого Лапласиана*. Данная функция изображения сходна с предыдущей с той разницей, что она подвергается операции гистограммного выравнивания с оригинальной функцией. Обозначим данную функцию как EALaplass.

c. *Аддитивное изображение с использованием сложного Лапласиана*. Усложнение заключается в том, что Лапласиан высчитывается не над оригинальным изображением, а над его размытием. Размытием изображения $f(x,y)$ будет результат свертки это изображения со следующим ядром:

$$S=\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix},$$

при этом фактор равняется 9. Вместо того, чтобы последовательно применять две операции свертки с двумя разными ядрами, эти ядра можно заменить одним равноценным ядром размерности 5×5:

$$E=\begin{pmatrix}-1&-2&-3&-2&-1\\-2&6&4&6&-2\\-3&4&1&4&-3\\-2&6&4&6&-2\\-1&-2&-3&-2&-1\end{pmatrix},$$

фактор равен 9. Тогда итоговое изображение будет определяться так:

$$h(x,y)=N(f(x,y)+N\mathrm{Conv}(f(x,y),M,x,y))).$$

Обозначим данную функцию как AELaplass.

d. *Выровненное аддитивное изображение с использованием сложного Лапласиана*. Как и в случае 2.b, результирующая функция выравнивается по гистограмме с оригинальным изображением. Обозначим данную функцию как EAELaplass.

3. Методы на основе выделения границ

a. *Оператор Собеля* [6]. Данный оператор использует две свертки изображения с ядрами

$$M_1=\begin{pmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{pmatrix},\ M_2=\begin{pmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{pmatrix}.$$

На основе полученных сверток G_1 и G_2 вычисляется общая свертка

$$G(x,y)=\sqrt{G_1(x,y)^2+G_2(x,y)^2}\ .$$

Полученная свертка вычитается из оригинального изображения: $h(x,y)=f(x,y)-G(x,y)$. Обозначим данную функцию как ASobel.

b. *Оператор Робертса* [7]. Метод основан на применении двух ядер

$$M_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ и } M_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix},$$

каждое из них имеет центр в точке (0,0). Конечное изображение определяется аналогично оператору Собеля. Обозначим данную функцию как ARobets.

c. *Оператор Прюитт* [8]. Данный оператор использует две свертки изображения с ядрами

$$M_1 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \; M_2 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

Конечное изображение определяется аналогично оператору Собеля. Обозначим данную функцию как APrewitt.

d. *Оператор Щарра* [6]. Данный оператор использует две свертки изображения с ядрами

$$M_1 = \begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}, \; M_2 = \begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}.$$

Конечное изображение определяется аналогично оператору Собеля. Обозначим данную функцию как AScarr.

4. Методы сглаживания изображения

a. *Простое сглаживание* [4, 12]. Метод основан на свертке изображения с ядром: $M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$, фактор равен 9. Итоговое изображение определяется как

$$h(x,y)=N\text{Conv}(f(x,y),M,x,y)).$$

Обозначим данную функцию как Smooth.

b. *Размытие по Гауссу* [4, 12]. Данный метод использует ядро: $M = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 1 \end{pmatrix}$, фактор равен 16. Итоговое изображение определяется как

$$h(x,y)=N\text{Conv}(f(x,y),M,x,y)).$$

Обозначим данную функцию как Gauss.

5. Метод медианного фильтра [5]. Для данного случая будем рассматривать окрестность точки (x,y) радиусом в 1 пиксель, представляющую собой множество из 8-ми пикселей, окружающих точку (x,y), а также содержащую саму точку (x,y). Обозначим окрестность как $U_f(x,y)$. Для каждой точки (x,y) изображения $f(x,y)$ находится медиана множества $U(x,y)$. Данное значение будет являться соответствующим значением пикселя (x,y) итогового изображения

$$g(x,y):\; g(x,y) = \text{median}(U_f(x,y)).$$

Обозначим данный метод как Median.

5 Автоматическое распознавание и коррекция ошибок

В предыдущих двух примерах количество распознанных слов и общее количество слов определялись вручную по мере просмотра каждого изображения в OCR. Однако для оценки качества распознавания больших коллекций данный метод не является рациональным. Кроме того, результирующий текст содержал в себе определенный процесс некорректно распознанных слов, т. е. таких слов, в которых присутствовало небольшое количество ошибок в символах (не больше 2). Человек вручную может легко «восстановить» оригинальное слово, однако это невозможно при больших объемах данных. Поэтому было решено разработать специальный текстовый анализатор, предназначенный для выявления и исправления слов с ошибками, а также для разделения всего текста на семантические составляющие, а именно: дата (день, месяц, год, а также слово, обозначающее дату) и отдельные предложения. Анализатор представляет собой скрипт на языке PHP. Алгоритм работы анализатора основан на применении функции поиска похожих слов. Ниже приведено описание данной функции:

1. За основу алгоритма взята функция, вычисляющая расстояние Левенштейна [9, 11] (минимальное количество замен, добавлений или удалений букв, для того чтобы из слова *A* получить слово *B*).
2. Алгоритм выполняет последовательное вычисления расстояний Левенштейна для данного слова, сравнивая его с каждым словом массива словоформ.
3. В результате алгоритм находит слово, которое ближе всего к данному.
4. В случае, когда минимальное расстояние Левенштейна среди всех слов массива превышает значение 3, алгоритм сообщает, что данное слово не распознано.
5. В случае, когда для слова, содержащего несколько ошибок, обнаруживается исправленная словоформа, оно считается исправленным и исправляется на эту словоформу.

Каждый файл, поданный на вход данному скрипту, обрабатывается следующим образом:

1. Задается массив слов-названий месяцев в 3 формах: в именительном, родительном падежах и в сокращении.
2. Задается массив слов русского языка, включая все словоформы. В качестве массива выступила база данных словоформ, найденная в интернете [1].
3. Задается массив всех возможных предлогов.
4. Задается массив всех возможных окончаний числительных.
5. В тексте находится ключевое слово, отвечающее за конец даты: «год» или «г.». Наличие одного из этих слов означает наличие даты в тексте. Часть текста, расположенная слева от данного ключевого слова, копируется в строку даты. Все, что справа, копируется в строку текста.

6. На основе двух самых распространенных форм записи даты: <день>, <месяц>, <год> и <месяц>, <день>, <год> выполняется поиск дня, месяца и года. Год и день определяются как числа, расположенные в определенном порядке относительно месяца. Поиск названия месяца осуществляется с помощью алгоритма поиска похожих слов, описанного выше.

7. Строка текста разбивается на предложения (разделитель предложений – точка). Каждое предложение разделяется на слова (разделитель слов – все возможные пробелы и пунктуация).

8. Для каждого слова в массиве словоформ с помощью вышеописанного метода находится соответствующая словоформа. Слово считается распознанным, если для него была найдена словоформа.

9. Отдельно выполняется поиск чисел, предлогов, числительных, символов номера, которые также считаются распознанными словами.

10. Слово, состоящее только из цифр, считается числом.

11. Если слово, предшествующее числу, состоит из одного – двух служебных символов либо является одним из символов N, №, #, то оно заменяется символом «№» и считается символом номера.

12. Слово, совпадающее с одним из слов массива предлогов, считается предлогом.

13. Слово, состоящее из числа, за которым следует окончание из массива окончаний числительных с возможным разделением дефисом, считается числительным.

14. Все остальные слова считаются мусором.

15. Полученная семантическая структура, содержащая информацию обо всех предложениях, словах и их типах, записывается в метафайл в определенном формате.

В процессе работы скрипта в памяти процесса накапливается информация обо всех предложениях, словах и типах слов обрабатываемого файла. Для удобства работы информация по всем файлам одного обрабатываемого альбома записывается в один файл. Данная информация образует следующую иерархическую структуру.

Файл:
- список метафайлов;

Метафайл:
- имя файла;
- индикатор даты – логическое значение, определяющее наличие даты в метафайле;
- день;
- месяц;
- год;
- список предложений;

Предложение:
- список слов;

Слово:
- слово;
- тип слова (слово, исправленное слово, предлог, числительное и др.);
- оригинальное слово – указывается только для исправленных слов с целью дальнейшей проверки правильности исправления.

Для дальнейшей обработки полученных метафайлов с целью определения качества распознавания, а также для организации альбомов с целью просмотра полученных результатов распознавания и сравнения их с оригинальными изображениями подписей была разработана специальная программа CaptionViewer. Программа предполагает создание альбомов фотографий на основе данных метафайла с целью дальнейшей обработки уже целых альбомов. Информация об альбомах и метафайлах представлена следующей базой данных:

Альбом:
- идентификатор альбома;
- название альбома;
- каталог изображений подписей;
- каталог текстовых файлов подписей;

Файл:
- идентификатор файла;
- идентификатор альбома;
- индикатор даты;
- день;
- месяц;
- год;
- метаинформация.

Данные о структуре отдельного файла было решено хранить в одном единственном поле за счет сериализации структур данных. Информация о содержащихся в файле предложениях и словах хранится в формате JSON [13]. Этот текстовый формат хранения объектов лучшим образом подходит для хранения иерархических данных в одном поле. Благодаря простоте и скорости сериализации/ десериализации данные метафайла пригодны для быстрой обработки.

6 Оценка качества распознавания, сравнение различных методов распознавания альбомов

Для сравнения результатов распознавания изображений, полученных различными методами улучшения, были выделены следующие признаки:

D_i – показатель определения даты файлов альбома *i*. Для каждого альбома рассчитывается суммарный коэффициент D_i^0. Для каждого файла альбома *i* коэффициент увеличивается на 1 при распознавании компонент даты:

a. ключевое слово конца даты («г.» или «год»);
b. год;
c. месяц;
d. день.

Далее D_i рассчитывается как $D_i^0/(4n_i)$, где n_i – количество файлов в альбоме.

ν_i – доля правильно распознанных слов для альбома *i*.

Пусть W_{ij}^o – количество слов в оригинальном текстовом файле *j* для альбома *i*. Предполагается, что изначально *n* файлов были распознаны разными методами, а результаты распознавания были сохранены в разных альбомах (отдельный альбом для каждого метода распознавания). Оригинальным файлом *i* для альбома *j* называется текстовый файл *i*, полученный распознаванием с помощью метода *j*.

Пусть W_{ij}^n – количество правильно распознанных слов в файле *i* для альбома *j*. Найдем $W_j^o = \max_i W_{ij}^o$ – наибольшее количество слов распознанного текста для файла *j* и максимальное общее количество слов оригинальных текстов: $W^o = \sum_j W_j^o$.

Тогда $\nu_i = \left(\sum_j W_{ij}^n \right) / W^o$.

ν_i^* – максимальная доля правильно распознанных слов среди файлов альбома *i*: $\nu_i^* = \max_j \nu_{ij}$, где $\nu_{ij} = W_{ij}^n / W_j^o$.

С помощью разработанной программы CaptionViewer было проведено сравнение методов предварительной обработки изображений, описанных в разделе 4. Полученные результаты анализа представлены в табл. 3. Значения критериев D_i, ν_i и ν_i^* указаны в процентах.

Таблица 3. Результаты текстового анализа различными методами. Словом Original был назван альбом, полученный прямым распознаванием выделенных подписей, т. е. без предварительного применения методов улучшения изображений

Имя альбома / метода	D_i	ν_i	ν_i^*
Cut	98	57	88
ALaplas	60	35	88
EALaplas	100	50	82
AELaplas	95	56	82
EAELaplas	98	56	93
ASobel	72	45	73
ARoberts	85	43	70
APrewitt	75	44	72
AScharr	82	53	80
Smooth	95	55	86
Gauss	92	54	82
Median	100	56	82
Original	90	52	77

Как видно из табл. 3, альбом Original уступает по указанным критериям многим из других альбомов, которые были получены с помощью улучшения изображений. К сожалению, добиться стопроцентного результата не удалось. Наилучшими методами оказались «Эвристический метод порогового отсечения без параметров», «Выравнивание аддитивного изображения с использованием сложного Лапласиана» и «Метод медианного фильтра».

7 Дальнейшие планы

1. *Семантический анализ подписи.* С помощью семантического анализа подписи предполагается выделять такие атрибуты, как место, сюжет и объект. Способ выделения времени фотографирования описан в разделе 5. Выделение остальных атрибутов предполагается осуществлять также с использованием словаря-тезауруса. Методы семантического анализа, основанные на использовании словаря-тезауруса, позволят определить в сообщении стилистически маркированную лексику, наличие которой в подписи отнесет ее к определенным атрибутам. Методы синтаксического анализа позволят определять в подписи те или иные синтаксические конструкции, типичные для каждого атрибута. Анализ контекстуального окружения части текста позволит установить дополнительные признаки того, принадлежит ли данная часть к определенному атрибуту или нет. Предполагается использование статистических методов анализа текста подписи. Такие методы используются, например, для выявления стилистических особенностей текста. Они предполагают построение вероятностной модели на основе анализа существующей выборки данных. Применение подобной модели позволит высчитывать для каждого части подписи некоторый коэффициент (соответствующий оценке вероятности того, что часть подписи принадлежит данному атрибуту) и идентифицировать его, если значение коэффициента выше некоторого установленного порогового значения.

Так, например, признаками места является наличие двух подряд идущих слов, первое относится к наименованию вида места (поселок, губа, деревня и т. д.). Второе относится к именам собственным (№ 4, Ведлозеро и т. д.), причем место фотографирования может быть задано одним или двумя значениями. Например, на представленной подписи – это поселок № 4 и губа озера Ведлозеро. При анализе подписей были использованы онтологии географических названий. Заметим, что на подписях были использованы некоторые сокращения географических названий, которые были включены в онтологический класс, например, Водл. вместо Водлозера, и др.

2. *Текстурный и контурный анализы.* С помощью методов анализа текстур участков фотографий предполагается выделять отдельные объекты: водную поверхность, стенки карьера, людей, флаги, растяжки с лозунгами и т. д. Сочетание текстурного и контурного анализов позволит выделять дома, машины, людей и т. д. Заметим, что текстурные характеристики существенно зависят от времени съемки (зима или лето).

Для решения задач поиска похожих текстур будут использоваться методы:

- фрактальная размерность Реньи [10];
- вейвлет-анализ [2];

- набор фильтров Габора [14].

Распознавание текстур ведется методом обучения с учителем. Выделяются обучающая и контрольная выборки. На фотографиях из обучающей выборки выделяем прямоугольные фрагменты разыскиваемого объекта размером $n \times m$ пикселей. Анализируя их, находим для рассматриваемого метода граничные параметры принадлежности текстур к типу разыскиваемых объектов. Обрабатывая контрольную выборку, проверяем работу алгоритма (правильность поиска).

Определение наличия людей на фотографиях проводится согласно следующего алгоритма. С помощью текстурного анализа находится участок с текстурой похожей на лицо человека, затем определяется контур этого участка, проводится его регуализация и контур сравнивается с возможными контурами лиц, учитывая наличие волос и головных уборов, т. е. обрезанный овал.

3. Планируется разработать программную систему для анализа фотографий, которая позволит указывать для каждой фотографии участки, соответствующие объектам, а также действия, совершаемые над ними. С помощью данной программы можно будет наделить каждую фотографию дополнительными сложными атрибутами.

4. Планируется разработать информационную систему поиска фотографий по атрибутам, полученным из подписи и анализа текстур. Таким образом, можно будет найти, например, все фотографии, на которых изображены карьер или какое-нибудь административное здание.

8 Заключение

В процессе решения поставленной задачи было сделано следующее:

- рассмотрены и опробованы различные способы распознавания больших объемов изображений;
- реализованы различные методы улучшения изображений;
- разработан механизм исправления ошибок распознанного текста;
- разработан формат хранения результатов распознавания;
- предложены различные критерии оценки качества распознавания текстов;
- создана программа, позволяющая работать с результатами распознавания и также сравнивать качество распознавания с помощью различных предложенных критериев.

Данная работа еще не закончена и будет продолжена в направлении семантического анализа подписей и текстурного и контурного анализа фотографий.

Литература

[1] Генерация всех словоформ (по мотивам словарей Ispell). – http://ispell.narod.ru/.

[2] Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.

[3] Дискретный оператор Лапласа. – Википедия. – http://ru.wikipedia.org/wiki/Дискретный_оператор_Лапласа.

[4] Каньковски П. Как работают фильтры размытия. – http://www.computerra.ru/gid/rtfm/graphic/35934/.

[5] Медианный фильтр. – Википедия. – http://ru.wikipedia.org/wiki/Медианный_фильтр.

[6] Оператор Собеля. – Википедия. – http://ru.wikipedia.org/wiki/Оператор_Собеля.

[7] Перекрестный оператор Робертса. – Википедия. – http://ru.wikipedia.org/wiki/Перекрёстный_оператор_Робертса.

[8] Прюитт. – Википедия. – http://ru.wikipedia.org/wiki/Прюитт.

[9] Расстояние Левенштейна. – Википедия. – http://ru.wikipedia.org/wiki/Расстояние_Левенштейна.

[10] Рогов А.А., Спиридонов К.Н. Применение спектра фрактальных размерностей Реньи как инварианта графического изображения // Вестник Санкт-Петербургского университета. Сер. 10. – 2008. – Вып. 2. – С. 30-43.

[11] Руководство по PHP. Levenshtein. – http://www.php.ru/manual/function.levenshtein.html.

[12] Image Processing for Dummies with C# and GDI+. Part 2. – Convolution Filters. – http://www.codeproject.com/KB/GDI-plus/csharpfilters.aspx.

[13] JSON. – Википедия. – http://ru.wikipedia.org/wiki/JSON.

[14] Movellan J.R. Tutorial on Gabor filters. – http://mplab.ucsd.edu/tutorials/gabor.pdf/

Analysis of typewritten captions in digital album

A.N. Talbonen, A.A. Rogov

This article is devoted to problems of creating a digital collection of photos from White Sea-Baltic Canal's construction. The article contains a description of the first step of carried work which aim was to read photos captions and to evaluate a quality of their recognition.