

АЛГОРИТМ АВТОМАТИЧЕСКОГО ВЫДЕЛЕНИЯ ПРИЗНАКОВ ДЛЯ РАСПОЗНАВАНИЯ СИМВОЛОВ

Аннотация. Статья содержит описание проблемы автоматического выделения признаков и предложение способа её решения.

Ключевые слова: выделение признаков, распознавание символов, нейронные сети, фильтрация, кластеризация.

Введение. Выделение признаков играет важную роль в области обработки изображений, в особенности, когда задачей алгоритмов является определение и изолирование различных интересующих частей или форм (признаков) в оцифрованном изображении или видеопотоке. К выделению признаков низкого уровня относят следующие алгоритмы: выделение границ (*edge detection*), углов (*corner detection*), областей (*blob detection*) [1]. Признаком называют часть исходных данных, являющуюся индивидуальным «свойством» представляющих наибольший интерес с точки зрения проблематики. Так, для эффективной работы алгоритма распознавания образов, выделение признаков предполагает собой определение минимального набора черт, требующихся для разграничения объекта одного класса от другого.

Проблематика. При работе с нейронными сетями такой топологии, как многослойный персептрон (нейроны в слоях которого реагируют на конкретные признаки или же группы скомбинированных признаков), применяя его при распознавании символов исследователи и непосредственно программисты сталкиваются с задачей ручного выделения признаков на изображениях, входящих в обучающую выборку. Так, для распознавания буквы «А» оптимально выделить 3 пересечения и две нижних оканчивающиеся линии (Рисунок 1). Можно определить следующий список аспектов проблематики:

- Имеется алфавит в виде множества изображений символов, которому обучается нейросеть;
- При обучении нейросети с учителем всегда требуется указать признаки у каждого из символов;
- В одном алфавите может быть как до 30 символов, так и значительно больше (иероглифические);
- Может потребоваться обучение сети нескольким алфавитам (японский + французский + русский).

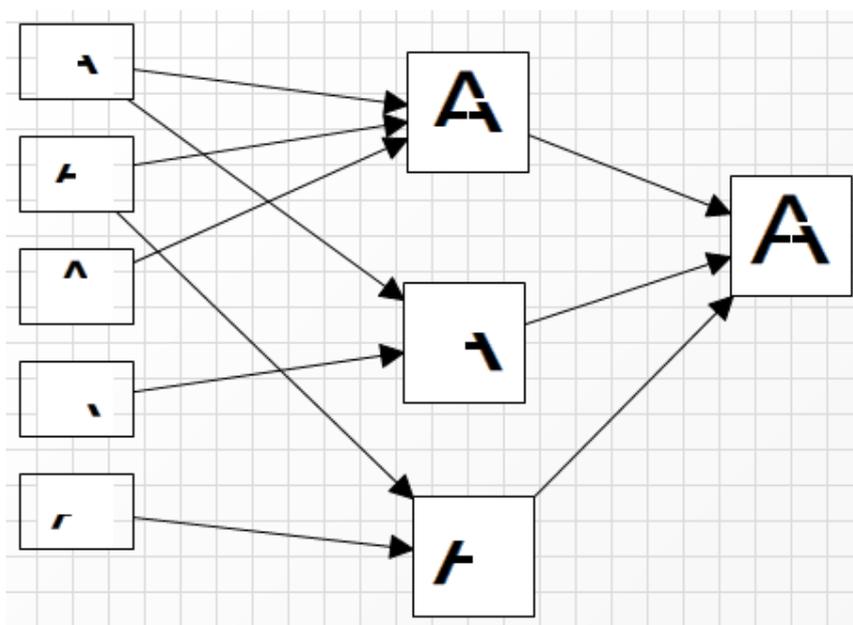


Рис. 1. Признаки из изображения буквы «А» в структуре многослойного персептрона

Исходя вышеуказанных условий, человеку при работе с многослойным персептроном (в случае обучения японскому, французскому и русскому одновременно) потребуется вручную выделить около **18 тыс.** признаков. Приняв среднее время ручного выделения каждого признака за 5 секунд, выполнение данной задачи вручную займет около **25 часов.**

Даже при рассмотрении менее критического случая обучения лишь русскому алфавиту (строчные и печатные буквы), потребуется 5 минут непрерывного выделения, что неприемлемо для программистов в ходе разработки и частых тестирований программных реализаций многослойного персептрона. При этом, любое малое изменение размеров букв в выборке может

свести на нет даже законспектированные координаты признаков в каждой букве.

Таким образом, целесообразно использование средства, позволяющего производить выделение признаков автоматически и при этом с точностью не ниже 90%, что оценивается последующим качеством распознавания нейронной сетью на изображениях тестовой выборки.

Обзор существующего алгоритма. Алгоритм, основанный на геометрической методике выделения признаков описан Dinesh Dileep в статье «*A Feature Extraction Technique Based On Character Geometry For Character Recognition*». Алгоритм основывается на «скелете» изображения символа и разделяет между собой линии (скопления пикселей черного цвета при однобитной палитре), которые образует конкретный символ [2].

Алгоритм состоит из следующих основных этапов:

- Предобработка (бинаризация, скелетизация)
Этап предобработки включает в себя бинаризацию (приведение палитры изображения к черно-белой, называемой однобитной), а также скелетизацию (популярный алгоритм скелетизации – скелетизация Зонга-Суня [3], пример на рисунке 4).
- Зонирование
Однобитное черно-белое изображение разбивается на 9 (3x3) одинаковых зон, к которым непосредственно применяется далее описанный алгоритм. Автор статьи отмечает такое разбиение оптимальным, т.к. искомый признак символа занимает свою конкретную зону в большинстве случаев [2].
- Нахождение стартовых, малых стартовых точек и пересечений
Производится обход по изображению с целью отнесения пикселей скелета символа к трем типам:

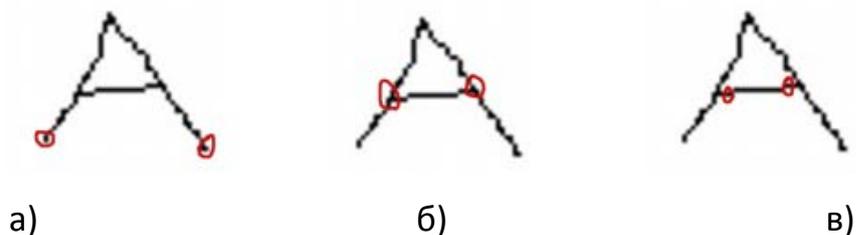


Рис. 2. Типы пикселей: стартовые точки, пересечения, малые стартовые точки.

Стартовые точки – пиксели с одним соседом (рядом находится лишь 1 пиксель черного цвета).

Пересечения - необходимо, чтобы пиксель как минимум имел более одного соседей. Соседние пиксели классифицируются на прямые и диагональные по отношению к конкретному пикселю в обходе; определение пересечения происходит по следующим критериям:

3 соседа – если ни один из пикселей не смежен друг с другом, пиксель является пересечением

4 соседа – если каждый прямой имеет в соседстве смежный диагональный пиксел (либо наоборот), пиксель не является пересечением

5 соседей – если пиксель имеет 5 и более соседей, он является пересечением

Малые стартовые точки – к данному типу могут быть отнесены пиксели с более чем 2 соседями. Если пиксель уже является пересечением, значит сегмент линии оканчивается здесь, и все соседи данного пикселя становятся малыми стартовыми точками.

Сначала определяются стартовые токи; затем обход стартует с первой стартовой точки, находя ближайшие пересечения; определив пересечение, сегмент линии (будущий признак) заканчивается и определяются малые стартовые точки.

- Разграничение сегментов

Полученные сегменты линий могут относиться к одному из типов линий:

горизонталь, вертикаль, правая диагональ, левая диагональ. Направление линии определяется цифрой (позицией) в соответствующей матрице, где С – текущий пиксель в обходе (рисунок 3). Например, для горизонтальной линии сегмент из трех пикселей может быть представлен числами «7, 7, 7».

4	5	6
3	С	7
2	1	8

Рис. 3. Матрица для определения типов линии.

- Если в сегменте линии больше всего пикселей с типом линии 2 или 6, то тип линии – правая диагональ;
- Если в сегменте линии больше всего пикселей с типом линии 4 или 8, то тип линии – левая диагональ;
- Если в сегменте линии больше всего пикселей с типом линии 1 или 5, то тип линии – вертикаль;
- Если в сегменте линии больше всего пикселей с типом линии 3 или 7, то тип линии – горизонталь.

Если два типа линии появляются одинаковое число раз, тип линии принимается от первой появившейся.

- Выделение признаков

Вектор признаков формируется после того, как сегменты отнесены к конкретному типу линии. Каждая из 9 зон в изображении содержит:

- Число сегментов-диагоналей, вертикалей и горизонталей
- Нормализованные длины по вышеуказанным сегментам

Номер каждой линии нормализуется по следующей формуле:

$$x = 1 - \left(\left(\frac{a}{10} \right) * 2 \right) (1),$$

Где а - общее число сегментов линий

Длина каждой линии нормализуется по следующей формуле:

$$x = \frac{b}{c} \quad (2),$$

Где b – число пикселей, относящихся к одному из типу линий;

c – число пикселей черного цвета в данной обрабатываемой зоне

Наконец, 9 зон с полученными данными по количеству сегментов и длинами представляют собой результат работы алгоритма и могут быть использованы в нейронной сети для распознавания символов [2].

Предложение способа решения. Использование алгоритма скелетизации в вышеописанном алгоритме несет с собой следующий отрицательный эффект: скелетизация серьезно упрощает изображение, при этом теряя некоторые особенности в индивидуальном начертании символов, такие как петли и засечки (рисунок 4).

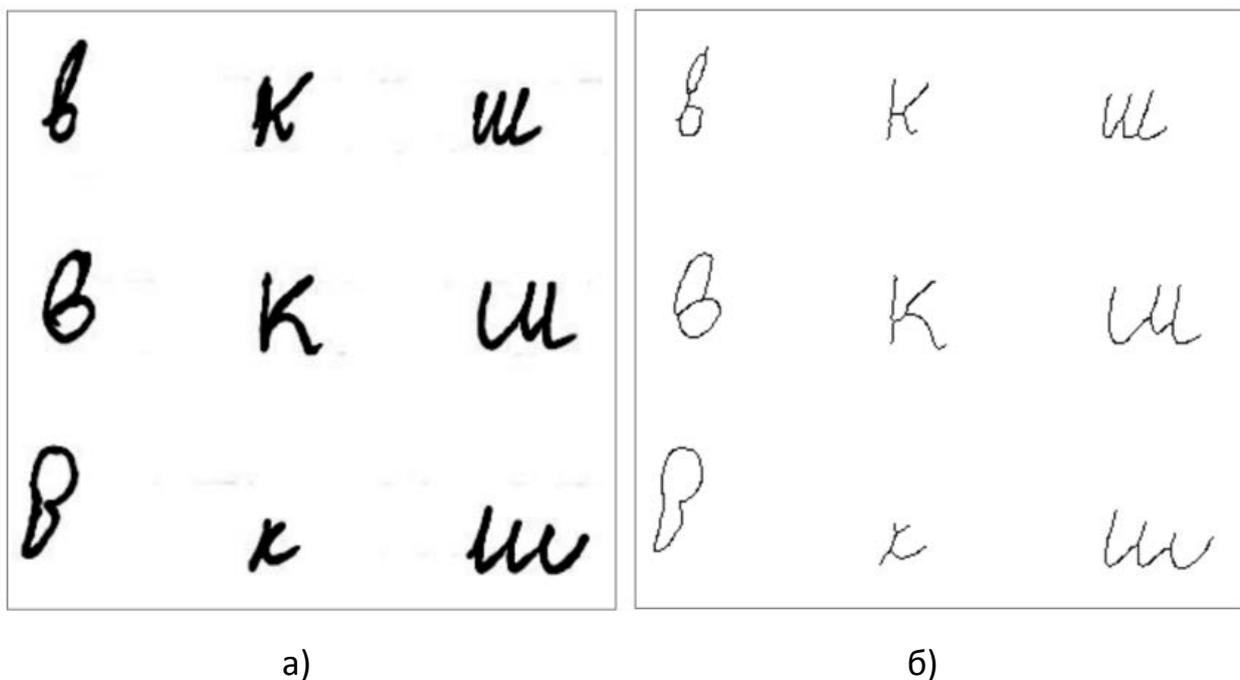


Рис. 4. Изображения символов букв до (а) и после (б) применения алгоритма скелетизации.

Поэтому, предлагаемый алгоритм несет в своей основе определенный фильтр, описание которого приведено ниже, и не использует скелетизацию.

Предлагаемый алгоритм выделения признаков содержит следующие этапы:

- Предобработка

Включает в себя увеличение яркости ~25% (формула 3) [4] и бинаризацию цветовой палитры изображения

$$x = x + N * \frac{128}{100} (3),$$

Где x – значение канала R, G или B;

N – процент увеличения яркости

- Фильтрация «сгустков»

Фильтр предполагает собой проход по всем пикселям черного цвета и призван оставить на однобитном изображении лишь скопления пикселей в точках символа, соответствующих координатам пересечений и оканчивающихся линий.

С каждым пикселем черного цвета ассоциировано значение x , представляющее собой величину «густоты». Изначально, $x=0$. Пусть d = шаг в градусах, причем 360 кратно d и $d \leq 90$. Тогда, каждый раз увеличивая шаг начиная с нуля до 360 прокладывается линия (рисунок 5) от конкретного пикселя в максимально дальний пиксель черного цвета, находящимся под углом, определяемым шагом. Величина «густоты» x увеличивается за счет длин всех проложенных линий. Таким образом, пиксели, находящиеся в пересечениях линий, будут иметь большее значение «густоты», нежели те, что преобладают посередине линий. Целесообразно увеличивать в несколько раз (варьируется ~1.0-2.0) каждое увеличение «густоты» x для пикселей, находящихся в оканчивающихся линиях (не более трёх пикселей-соседей черного цвета), таким образом учитывая края оканчивающихся линий при фильтрации. Пиксели с x ниже ~75% от максимального отфильтровываются.

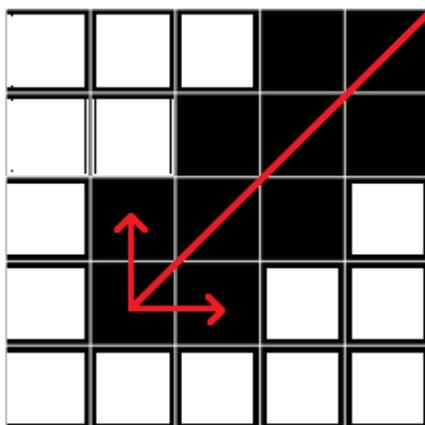


Рис. 5. Пример прокладывания линий с шагом в 45 градусов.

- Кластеризация пикселей

После завершения фильтрации «сгустков», данные скопления пикселей должны быть сведены к отдельным кластерам, координаты центроидов которых послужат координатами местонахождения признака символа (рисунок 6).



Рис. 6. Пример фильтрации «сгустков» для буквы «А».

Для реализации кластеризации целесообразно использовать алгоритм с непредопределенным заранее числом кластеров, которым является алгоритм g-means. Алгоритм представляет собой усовершенствование k-means с использованием проверки на нормальное распределение. Начиная с числа кластеров $k=2$, запускается стандартный k-means и определяются первые два кластера-кандидата. Между кандидатами проводится вектор, на который проецируются точки (рисунок 7) [5]. При несоответствии распределения нормальному по критерию Андерсона-Дарлинга,

описанным в статье «*Learning The K In K-Means*» [6], действия повторяются и создаются новые кандидаты до тех пор, пока распределение всех кластеров не будет соответствовать нормальному.

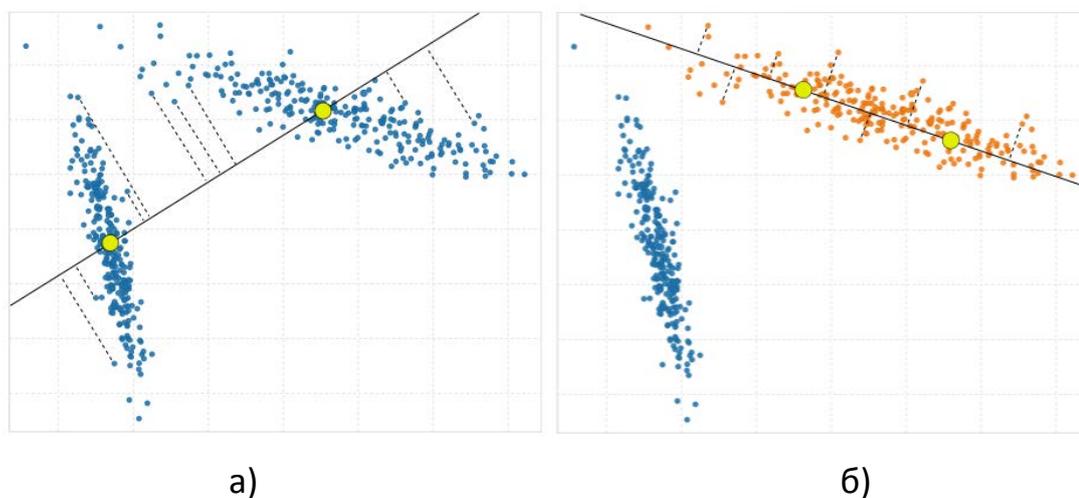


Рис. 7. Иллюстрация алгоритма кластеризации g-means.

Таким образом, полученные координаты центроидов кластеров возможно использовать в качестве координат признаков, используемых в нейронной сети.

Заключение. В статье была рассмотрена проблема автоматического выделения признаков. Описан существующий способ решения и выдвинут оригинальный способ, учитывающий аспекты проблематики.

СПИСОК ЛИТЕРАТУРЫ

1. Wikipedia [Электронный ресурс]: Feature extraction URL: https://en.wikipedia.org/wiki/Feature_extraction (дата обращения: 10.04.2017).
2. Dinesh Dileep A feature extraction technique based on character geometry for character recognition // Cornell University Library URL: <https://arxiv.org/pdf/1202.3884v1.pdf> (Дата обращения: 12.04.2017).
3. Quiet place - Алгоритм скелетизации Зонга-Суня (Zhang-Suen) [Электронный ресурс] URL: http://corwal.blogspot.ru/2013/11/blog-post_20.html (Дата обращения: 13.04.2017).

4. Программирование на C, C#, Java [Электронный ресурс] : Фильтрация изображений на C# URL: <http://vscode.ru/prog-lessons/filtratsiya-izobrazheniy-na-c-sharp.html> (дата обращения: 09.04.2017).
5. bigml [Электронный ресурс] : Divining the 'K' in K-means Clustering URL: <https://blog.bigml.com/2015/02/24/divining-the-k-in-k-means-clustering/> (дата обращения: 10.04.2017).
6. Greg Hamerly, Charles Elkan Learning the k in k-means // Department of Computer Science and Engineering, University of California, San Diego URL: <http://papers.nips.cc/paper/2526-learning-the-k-in-k-means.pdf> (Дата обращения: 14.04.2017).